MAT-90-15

Upgrading of DART Software to 386 Computer Specifications

Ontario
Ministry
of
Transportation

Research and
Development
Branch

Ontario

Ministry of Transportation

Research and Development Branch

Documentation Page MAT-90-15

Upgrading of DART Software to 386 Computer Specifications

Author(s): T. Chung - Senior Research Engineer
C.R. Carter - Professor, McMaster University

Date Published: December, 1990

Published by: Research and Development Branch, MTO

Contact Person: Tony Masliwec, Materials Office, Research and Development Branch
(416) 235- 4689

Abstract: The existing DART prototype unit has proven to be of considerable value for the investigation of concrete bridge decks, especially those with a bituminous surfacing. However, the unit is based on 1982 technology and the data processing device is an XT-computer which has inadequate memory and takes several hours to process data from an average sized bridge deck. Attempts to privatize the DART technology and produce a commercial unit in the private sector have been unsuccessful. In order to take advantage of the technology, it is necessary to upgrade the prototype unit.

In this report, a 386-based computer, with increased computer memory size, increased disk data storage and increased speed of data processing, is used to replace the XT-based DART. The 386-based DART is demonstrated to have greatly improved operational efficiency and effectiveness. The procedures for data acquisition to be made directly by computer, without the need for an expensive analog data recorder, are implemented. The existing software is revised to handle the new input/output format, data storage, processing, and interface devices. The 386-based DART has been field tested at the Ninth Line and Bobcaygeon bridge decks. Data are collected and processed using the new system configuration. The processing capability of the 386-based DART is noted to be 25 times faster than its predecessor. Also, the processed results are compared with surveys conducted by Penetradar. It is shown that both processing strategies offer similar performance.

Comments: Final report of Research and Development Branch project 22212

Key Words: impulse radar, deck assessment, structures, durability, software

Distribution: Internal and on request

Copyright Status: Crown copyright © 1990 Ministry of Transportation

MAT-90-15

Upgrading of DART Software to 386 Computer Specifications

T. Chung
Senior Research Engineer

C.R. Carter
Professor

Communications Research Laboratory
McMaster University
Hamilton

Published by
The Research and Development Branch
Ontario Ministry of Transportation

Published without prejudice as to the application of the findings. Crown copyright reserved; however, this document may be reproduced for non-commercial purposes with attribution to the Ministry.

For additional copies, contact:
The Editor, Technical Publications
Room 320, Central Building
1201 Wilson Avenue
Downsview, Ontario
Canada M3M 1J8

Telephone: (416) 235-3480
Fax: (416) 235-4872

December 1990

Table of Contents

Executive Summary viii

1/ Introduction 1

1.1/ Background 1

1.2/ Objectives 1

1.3/ Overview 3

2/ 386-DART System Configuration 4

2.1/ Criteria for Selecting the New Computer System 4

2.1.1/ System Memory 4

2.1.1.1/ Current XT-Based DART 4

2.1.1.2/ Proposed 386-Based DART 5

2.1.2/ System Speed 6

2.1.2.1/Current XT-Based DART 6

2.1.2.2/Proposed 386-Based DART 6

2.1.3/ Storage Devices 7

2.1.3.1/Current XT-Based DART 7

2.1.3.2/Proposed 386-Based DART 7

2.1.4/ Data Acquisition Interface Card 7

2.1.4.1/Current XT-Based DART 7

2.1.4.2/Proposed 386-Based DART 7

2.1.5/ System Durability 8

2.2/ Generic Specification for 386-Based DART 9

2.3/ Data Acquisition Sub-System for 386-Based DART 10

2.4/ System Configuration Design for 386-Based DART 11

2.5/ Summary 12

3/ The 386-Based Software Algorithms 14

3.1/ Main Program - DART90 14

3.1.1/ Subroutine FPLATE90 16

3.1.2/ Subroutine REMIR90 17

3.1.3/ Subroutine SIGNAL90 17

3.1.4/ Subroutine PEAKCD90 20

3.1.5/ Subroutine PEAKS90 20

3.1.6/ Subroutine TABLE90 21

3.2/ Output Program - DECK90 21

Table of Contents (continued)

3.3/ Summary 21

4/ Processing of the Ninth Line Bridge Deck Data 23

4.1/ Survey Using the XT-Based DART 23

4.2/ Survey Using the 386-Based DART 24

4.2.1/ Bridge Deck Profile 25

4.2.2/ Waveforms Analysis 26

4.2.2.1/ Thickness Measurements 26

4.2.2.2/ Faults Identification 27

4.2.3/ Bridge Deck Assessment 28

4.3/ Comparison with Penetradar's Results 30

4.4/ Summary 32

5/ Processing of the Bobcaygeon Bridge Decks Data 33

5.1/ Assessment of the Little Bob South Bridge 33

5.1.1/ Bridge Profile Plots 33

5.1.2/ Radar Inspection 34

5.1.3/ Comparison with Penetradar's Results 35

5.2/ Assessment of the Little Bob North Bridge 36

5.2.1/ Results from 386-Based DART 37

5.2.2/ Comparison with Penetradar's Results 38

5.3/ Summary 39

6/ Conclusions and Recommendations for Further Research 41

6.1/ Conclusions 41

6.2/ Recommendations for Further Research 45

7/ References 47

8/ Tables 49

9/ Figures 59

Appendix 145

List of Tables

1/ Survey summary for Ninth Line evaluated by XT-based DART (1988) 49
2/ Survey summary for Ninth Line evaluated by 386-based DART 50
3/ Survey for Ninth Line from Penetradar's radar inspection 51
4/ Survey summary for Little Bob South Bridge 53
5/ Survey for Little Bob South Bridge from Penetradar's radar inspection 54
6/ Survey summary for Little Bob North Bridge 56
7/ Survey for Little Bob North Bridge from Penetradar's radar inspection 57

List of Figures

1/ System configuration design for 386-based DART 59
2/ Flow diagram for main program DART90 60
3/ Sample inputs for DART90 software 61
4/ (a) Superimposed waveforms from flat plate signal and internal antenna reflections. 62
(b) Corrected flat plate signal 62
5/ Flow diagram for subroutine FPLATE90 63
6/ Flow diagram for subroutine REMIR90 65
7/ Flow diagram for subroutine SIGNAL90 66
8/ Flow diagram for subroutine PEAKCD90 68
9/ Flow diagram for subroutine PEAKS90 69
10/ Flow diagram for subroutine TABLE90 70
11/ Flow diagram for program DECK90 71
12/ Delaminated areas at Ninth Line detected by XT-DART (1988) 72
13/ Debonded areas at Ninth Line detected by XT-DART (1988) 73
14/ Sample waveforms from Ninth Line using XT-DART (1988) 74
15/ Waveforms from Line 3 of Ninth Line using XT-DART (1988) 75
16/ Waveforms from Line 5 of Ninth Line using XT-DART (1988) 76
17/ Waveforms from Line 1 of Ninth Line sampled by 386-DART 77
18/ Waveforms from Line 2 of Ninth Line sampled by 386-DART 78
19/ Waveforms from Line 3 of Ninth Line sampled by 386-DART 79
20/ Waveforms from Line 4 of Ninth Line sampled by 386-DART 80
21/ Waveforms from Line 5 of Ninth Line sampled by 386-DART 81
22/ Waveforms from Line 6 of Ninth Line sampled by 386-DART 82
23/ Waveforms from Line 7 of Ninth Line sampled by 386-DART 83
24/ Waveforms from Line 8 of Ninth Line sampled by 386-DART 84
25/ Waveforms from Line 9 of Ninth Line sampled by 386-DART 85
26/ Waveforms from Line 10 of Ninth Line sampled by 386-DART 86
27/ Ninth Line: peaks and covers measurement
(a) Line 1, (b) Line 2 87
28/ Ninth Line: peaks and covers measurement
(a) Line 3, (b) Line 4 88
29/ Ninth Line: peaks and covers measurement
(a) Line 5, (b) Line 6 89

List of Figures (continued)

30/ Ninth Line: peaks and covers measurement
(a) Line 7, (b) Line 8 90
31/ Ninth Line: peaks and covers measurement
(a) Line 9, (b) Line 10 91
32/ Ninth Line: 'rone' and 'rtwo' ratios (a) Line 1, (b) Line 2 92
33/ Ninth Line: 'rone' and 'rtwo' ratios (a) Line 3, (b) Line 4 93
34/ Ninth Line: 'rone' and 'rtwo' ratios (a) Line 5, (b) Line 6 94
35/ Ninth Line: 'rone' and 'rtwo' ratios (a) Line 7, (b) Line 8 95
36/ Ninth Line: 'rone' and 'rtwo' ratios (a) Line 9, (b) Line 10 96
37/ Scaled areas at Ninth Line detected by 386-DART 97
38/ Debonded areas at Ninth Line detected by 386-DART 98
39/ Delaminated areas at Ninth Line detected by 386-DART 99
40/ Ninth Line waveforms from Line 4:
(a) at 12 m indicates good condition. (b) at 18 m indicates good condition. 100
(c) at 30 m indicates good condition. (d) at 40 m indicates good condition 101
41/ Ninth Line waveforms from Line 5:
(a) at 12 m indicates delamination. (b) at 18 m indicates debonding. 102
(c) at 30 m indicates delamination. (d) at 40 m indicates scaling 103
42/ Ninth Line waveforms from Line 6:
(a) at 25 m indicates delamination. (b) at 35 m indicates delamination. 104
(c) at 39 m indicates scaling. (d) at 45 m indicates debonding 105
43/ Delaminated areas at Ninth Line detected by Penetradar 106
44/ Debonded areas at Ninth Line detected by Penetradar 107
45/ Waveforms from Line 1 of Little Bob South Bridge 108
46/ Waveforms from Line 2 of Little Bob South Bridge 109
47/ Waveforms from Line 3 of Little Bob South Bridge 110
48/ Waveforms from Line 4 of Little Bob South Bridge 111
49/ Waveforms from Line 5 of Little Bob South Bridge 112
50/ Waveforms from Line 6 of Little Bob South Bridge 113
51/ Waveforms from Line 7 of Little Bob South Bridge 114
52/ Waveforms from Line 8 of Little Bob South Bridge 115
53/ Waveforms from Line 9 of Little Bob South Bridge 116
54/ Waveforms from Line 10 of Little Bob South Bridge 117
55/ Detailed profile for Lines 3 and 4 of Little Bob South Bridge 118

List of Figures (continued)

56/ Scaled areas at South Bridge detected by 386-DART 119

57/ Debonded areas at South Bridge detected by 386-DART 120

58/ Delaminated areas at South Bridge detected by 386-DART 121

59/ Little Bob South Bridge waveforms indicate good condition
(a) Line 2 at 8.5 m, (b) Line 7 at 9 m 122

60/ Little Bob South Bridge waveforms indicate delamination
(a) Line 3 at 10 m, (b) Line 8 at 12 m 123

61/ Little Bob South Bridge waveforms indicate debonding
(a) Line 7 at 3.5 m, (b) Line 9 at 17 m 124

62/ Little Bob South Bridge waveforms indicate scaling
(a) Line 1 at 4.5 m, (b) Line 5 at 1.6 m 125

63/ Delaminated and scaled areas at Little Bob South Bridge detected by Penetradar 126

64/ Waveforms from Line 1 of Little Bob North Bridge 127

65/ Waveforms from Line 2 of Little Bob North Bridge 128

66/ Waveforms from Line 3 of Little Bob North Bridge 129

67/ Waveforms from Line 4 of Little Bob North Bridge 130

68/ Waveforms from Line 5 of Little Bob North Bridge 131

69/ Waveforms from Line 6 of Little Bob North Bridge 132

70/ Waveforms from Line 7 of Little Bob North Bridge 133

71/ Waveforms from Line 8 of Little Bob North Bridge 134

72/ Waveforms from Line 9 of Little Bob North Bridge 135

73/ Waveforms from Line 10 of Little Bob North Bridge 136

74/ Scaled areas at North Bridge detected by 386-DART 137

75/ Debonded areas at North Bridge detected by 386-DART 138

76/ Delaminated areas at North Bridge detected by 386-DART 139

77/ Little Bob North Bridge waveforms indicate good condition
(a) Line 3 at 12 m, (b) Line 4 at 10 m 140

78/ Little Bob North Bridge waveforms indicate delamination
(a) Line 4 at 15 m, (b) Line 6 at 12 m 141

79/ Little Bob North Bridge waveforms indicate debonding
(a)Line 4 at 4 m, (b) Line 5 at 18 m 142

80/ Little Bob North Bridge waveforms indicate scaling
(a) Line 1 at 5 m, (b) Line 2 at 4 m 143

81/ Delaminated and scaled areas at Little Bob North Bridge detected by Penetradar 144

Executive Summary

Background

The MTO DART system has enjoyed considerable success in its research phase in evaluating concrete bridge decks by providing accurate information on the thickness of asphalt, the concrete cover over reinforcement and the identification and location of damage on bridge decks. The system has become operational and it is recognized that improvement can be made to increase efficiency, improve effectiveness and reduce time required to collect and process data. The benefits of these improvements lead to reduced overall costs.

Replacing the DART's XT-computer with a 386-based computer will allow for increased memory and speed of data processing. This will also allow data acquisition to be made directly by computer and eliminate the need for a high level magnetic data recorder costing approximately $50k. The existing algorithms and codes would remain valid. Software would require modification for the revised input/output format, data storage, handling, processing, and interface devices.

Objectives

The primary objectives of the present project include:

1. Replacing the existing XT-based DART with a 386-based computer.

2. Configuring the 386-based DART with increased computer memory size, increased disk data storage and increased speed of data processing.

3. Investigating the possibilities of eliminating tape recorded analog data by

improving data acquisition methods.

4. Upgrading the existing signal processing algorithms to enhance the full capabilities of the 386-based computer.

5. Evaluating the new system in the field on a known deck, for example the Ninth Line bridge deck in Oakville.

6. Testing the new system for a selected number of bridge decks to ensure that the system operates satisfactorily.

7. Comparing the survey results obtained above with the Penetradar's processing strategy.

Summary of Results

1. The XT-based computer, of the existing DART prototype unit, is replaced by a 386-based computer which includes hardware such as: 1) an industrial design 386-computer with 20 MHz CPU; 2) a 16 Mb system memory; 3) a 20 MHz 80387 math-coprocessor; 4) a 150 Mb hard drive; 5) one 5-1/4" 360 Kb and one 3-1/2" 1.44 Mb floppy drives; 6) one DT2821 A/D interface card; 7) one HP 7475A plotter; and 8) one Fujitsu DL2600 printer.

2. The upgraded system eliminates the need for tape recording analog signals. It is due to the fact that with greatly increased processing speed, larger system memory and hard disk capacity, an entire pass of DART data can be digitized directly, through the interactions of the DT2821 A/D card and ATLAB software, into computer memory. Upon the completion of data collection, the digitized data are retrieved from the extended memory to the system memory for waveform analysis.

3. The signal processing algorithms for the XT-based DART have been revised to exploit the full capacity of the 386-based computer system. The new software allows radar waveforms to be sampled once every 10 cm, instead of once every 20 cm.

4. Using the new system, data were collected and processed from the Ninth Line bridge deck in Oakville; Little Bob South bridge and North bridge in Bobcaygeon. It was found that the new system is 25 times faster in processing the data than its predecessor.

5. Survey results obtained from DART were compared with Penetradar. It was found that:

i) Both results agreed well on evaluating the Ninth Line bridge deck. There is +/- 3.61% difference in measuring the overall asphalt cover and +/- 1.28% difference in measuring the overall concrete cover. DART identified 2.74% scaling; 3.80% delamination and 3.53% debonding. Penetradar identified no scaling; 3.20% delamination and 9.40% debonding. Both surveys indicated that delamination was mostly on the south bound lane. The areas where Penetradar found debonding were also identified by DART. The scaled areas found by DART are clustered around the same localities where debonding are detected. It is likely that the scaled areas may actually be debonding where trapped moisture gives radar signatures that resemble scaling. In this case, the magnitude of the radar reflections, at peaks B and C, are greatly increased. Consequently, the ratios 'Rone' and 'Rtwo' exceed the threshold levels which are set for identifying scaling.

ii) For the Little Bob South bridge deck, the two surveys yielded +/- 13.03% difference in measuring the overall average asphalt and +/- 3.52% difference in measuring the overall concrete cover. DART identified 21.15% scaling; 28.90% delamination and 2.77% debonding. Penetradar identified 21.46% scaling; 18.96%

delamination and no debonding. Both surveys showed good agreement on locating scaled areas. DART found that delamination was significant at the centre section of the bridge deck. Radar waveforms from these areas exhibited badly defined 'W-characteristic' feature.

iii) For the Little Bob North bridge deck,. the two surveys yielded +/- 9.57% difference in measuring the overall average asphalt and +/- 4.27% difference in measuring the overall concrete cover. DART identified 25.51% scaling; 14.40% delamination and 5.58% debonding. Penetradar identified 26.04% scaling; 20.00% delamination and no debonding. Again, both surveys had good agreement on identifying scaled areas.

6. By comparing the surveys obtained from DART and Penetradar, it was noted that DART sampled waveforms once every 10 cm whereas Penetradar sampled waveforms once every 50 cm. Statistically, DART provides a greater minimum estimate on the condition of the bridge deck, since it covers more areas than Penetradar. Thus, on a badly deteriorated bridge deck, DART would provide a larger percentage of deterioration in comparing with Penetradar's results.

7. The discrepancies, found in calculating the asphalt thickness for the two bridge decks at Bobcaygeon, between the DART's results and Penetradar's results were due to the differences in the signal processing strategies. DART corrected each radar waveform by subtracting the antenna internal reflections; and, the dielectric constant of asphalt was calculated for each waveform. Penetradar did not subtract the antenna internal reflections in their signal analysis; and, the dielectric constant of asphalt was calculated once using core sample information.

8. The areas where DART claimed debonding for the two bridge decks at Bobcaygeon may actually be scaling judged from the age of the bridge deck. The radar waveforms from these areas showed very prominent reflection at peak C and

less prominent reflection at peak B. It is noted that DART employs two ratios, 'Rone' (which is the magnitude at peak B to the magnitude at peak A) and 'Rtwo' (which is the magnitude at peak C to the magnitude at peak A), to differentiate scaling from debonding; whereas, Penetradar only measures the increase in magnitude at peak C for determining scaling and debonding. Therefore, DART is more sensitive, in comparison to Penetradar, in terms of identifying scaling and debonding.

1/ Introduction

1.1/ Background

The DART has proven of considerable value for assessing concrete bridge decks, especially those with a bituminous surfacing [1-10]. The system employs an XT-based computer for performing signal analysis. With increasing demand from the present requirements of bridge deck assessment, it is believed that the XT-based computer is incapable of handling the work load. The major shortcomings of the XT-based DART are slow; memory deficiency; and lack of storage capacity. In order to increase the effectiveness of DART's operation, it is necessary to upgrade the computer unit.

A 386-based computer is selected to replace the XT-based computer. The new computer is configured to have faster clock speed; increased system memory and storage capacity than its predecessor. It is noted that the existing data acquisition procedures and processing algorithms would require modification for the new hardware.

1.2/ Objectives

The project objectives are two-fold:

1. Modify the existing DART system to increase the operational efficiency and effectiveness by:

a) incorporating a 386-based computer with increased computer memory size, increased disk data storage, and increased speed of data processing.

b) eliminating the need to tape record the analog waveforms. This can be accomplished by providing on-site computing and evaluation. These improvements would fully exploit the capabilities of the 386-based computer system. Further, since the tape recorder is eliminated, it is important to provide a means of storage of computer processed data using floppy disks.

c) upgrading the existing software to enhance the capabilities of a 386-based system recognizing that the existing signal processing algorithms and codes should still remain valid. This will require software modifications which are compatible with the existing packages and written in Fortran 77.

d) re-examining the interfacing hardware in order to incorporate any reductions in raw data input thus reducing computer storage requirements.

2. Evaluate and field test the new system by:

a) using the new system to collect and process bridge deck data from the Ninth Line bridge deck in Oakville in order to ensure that the new system operates satisfactorily in the field on a known deck.

b) using the data set from the Ninth Line bridge deck, a comparison of the new 386 DART software will be made with that of the old DART's software. As well, a comparison is to be made with Penetradar's data processing capability.

c) interfacing with MTO personnel on-site for a selected number of bridge decks to ensure successful transfer of technology for the new system. The analysis should also be compared with Penetradar's radar system.

1.3/ Overview

In chapter Two, a generic specification for a 386-based computer system is examined. The exact computer configuration requirements and the interface hardware are determined. Also, the practicability of eliminating the tape recorder and how this change impacts overall operations are studied. Finally, a system configuration design for DART is presented along with some technical information.

Chapter Three discusses the development of the 386-based software package. In general, the codes take advantage of the processing capabilities of the new computer and are compatible with the XT-based system. However, the data acquisition algorithm has to be revised to work with the new A/D interfacing card. Operation procedures and flow diagrams of each program module are given to assist DART's user to understand the software.

Chapter Four provides the survey for the Ninth Line bridge deck in Oakville. Data are collected and processed using the new system. The results are compared with the previous assessment obtained by the XT-based DART. This step ensures that the new system is functioning properly and operates satisfactorily in the field on a known deck. In addition, the results are compared with Penetradar's findings.

Chapter Five provides assessment for two bridge decks at Bobcaygeon. It is seen that the 386-based DART system is more operationally efficient, and yet compatible, in comparing with its predecessor. The results are also compared with Penetradar's system.

Chapter Six provides conclusions and recommendations for further research.

2/ 386-DART System Configuration

The DART project has enjoyed considerable success over the years by providing an accurate assessment of asphalt covered bridge decks. However, with the existing XT-computer, the throughput rate is extremely slow. The system has reached the operational stage and it is recognized that improvements can be made to increase efficiency, improve effectiveness and reduce time required to collect and process data. Therefore, it is necessary to upgrade the microcomputer system. A 386-based computer is shown to meet the above requirements.

There are few important criteria, in order to bring the DART project to full capabilities, which have been considered prior to selecting a 386-based computer system. The criteria include: 1) system memory; 2) system speed; 3) storage devices; 4) data acquisition sub-system and 5) system durability. These issues are discussed in the next section. As there is a large array of computer hardware available in the market, it is considered that a generic specification for the new computer system is adequate. Finally, a system configuration design for DART is given and some technical information is also provided.

2.1/ Criteria for Selecting the New Computer System

2.1.1/ System Memory

2.1.1.1/ Current XT-Based DART -- In the XT-based system, the system memory (RAM) is only 640 kilobytes (Kb). This is a very limited capacity for the DART project. Since the system overhead and the DART's software easily consume several hundreds of Kb of memory, there is only a small amount of RAM available for performing data acquisition. Thus, the system becomes ineffective if bridge decks of a hundred metres length or more are encountered. Previously, DART tape recorded analog waveforms at every bridge deck. The tape recorded data were analyzed off-site by playing the data into a computer using the current DART's software.

The existing DART's software can only sample a maximum of 225 Kb of tape recorded data into the system. One complete radar waveform requires 1 Kb of memory. Since the fifth wheel is equipped to sample radar waveforms every 10 cm, only 225 waveforms (or 22.5 m of bridge length) can be processed. This is a rather limited range for performing bridge deck assessment. In order to cover a longer distance, the software has been adjusted to sample data every 20 cm. As a result the system can process waveforms covering 45 m of bridge length at the expense of losing resolution.

Furthermore, for bridge decks over 45 m in length, the signal processing procedure is performed in a segmentation fashion (i.e. sampling 45 m of data once every 20 cm from the tape to system memory then stopping the recorder, transferring the data to hard disk and performing signal processing). In order to complete the evaluation of one pass of a 1 Km bridge deck, this procedure must be repeated 23 times. This is very ineffective, since on many occasions DART has to deal with multiple lane bridge deck of hundreds of metres of length.

2.1.1.2/ Proposed 386-Based DART -- Current technology allows hundreds of Megabytes (Mb) of system memory for microcomputers. It is believed that with this mass storage, the need for tape recording analog radar signals is redundant. Assuming that an appropriate data acquisition interface card is used for the new computer, data collection can be achieved by directly sampling radar signals into the computer memory. Once data are stored in the computer, they can be easily backed up onto floppy diskettes if desired. Since, the data are in digital format they are not vulnerable to degradation as opposed to the tape recorded analog signals. From past experience, it has been found that the quality of the cassette tapes deteriorate substantially after repeated times of tape play-back, thus reducing the effectiveness and accuracy of the system.

For the DART project, 1 Km of bridge deck data sampled once every 10 cm requires 10 Mb of memory. In order to eliminate tape recording data, the new system requires at least 12 Mb RAM with the extra 2 Mb for overhead. It is believed that this memory space should accommodate enough data storage for the majority of bridge decks in the province.

2.1.2/ System Speed

2.1.2.1/ Current XT-Based DART -- The XT-system is an 8088-based machine. The speed of this CPU is only 4.77 MHz. This is an extremely slow speed for most serious applications, such as the DART. Although the system is equipped with an 8087 math coprocessor, it can only speed up the numerical processing but does nothing to reduce the overhead time. For example, it requires three minutes to retrieve 225 Kb binary data from the hard disk and two seconds to process one waveform using the existing hardware. The total time required to process one complete pass of a 500 m long bridge deck is approximately two hours. For a six-passes on a 500 m long bridge deck, 12 hours of processing time are needed.

2.1.2.2/ Proposed 386-Based DART -- The 386-based machines run at a number of different speeds including: 16, 20, 25 or 33 MHz. The DART project requires a system having the combination of high speed, reliable operation and ease of maintenance. The 20 MHz machine satisfies all of these requirements since the basic clock speed is 4 times faster than the XT and the throughput rate is estimated to be at least 10 times higher than the existing system. For one pass on a 500 m long bridge deck, the processing time is less than 12 minutes; and, for evaluating a six-passes on a 500 m long bridge deck the overall processing time can be reduced to less than 1.5 hours. Thus, a 386-based machine with at least a 20 MHz CPU and a 80387 math coprocessor are recommended for the DART.

2.1.3/ Storage Devices

2.1.3.1/ Current XT-Based DART -- The XT-machine initially purchased (in 1983) for the DART project had only 10 Mb of hard disk capacity, but two years later it was upgraded to 20 Mb. Presently, the 20 Mb storage capacity is marginal since most of the disk space has been occupied by the DART's software package, operating system utilities and software libraries etc. There is only a limited amount of disk space left for the DART's data storage.

The floppy drive can access the 360 Kb diskette which has a data transfer rate of only 250 K bits/sec. With a high-density (e.g. 1.44 Mb) floppy disk, a faster speed (500 K bits/sec) and bigger storage size can be achieved.

2.1.3.2/ Proposed 386-based DART -- Hard disks come with many sizes such as 40, 70, 140 Mb and above. For the DART project, bridge decks with spans of hundreds of metres of length require a large disk space storage for raw radar waveforms. Therefore, a hard disk of about 100 Mb is recommended. (Note that 100 Mb can store 10 Km of radar waveforms). A 1.44 Mb high-density floppy disk drive is needed for the new system with the 360 Kb floppy disk drive as a second drive.

2.1.4/ Data Acquisition Interface Card

2.1.4.1/ Current XT-Based DART -- The interface card (model LAB MASTER manufactured by TECMAR INC) installed in the XT-system was designed to communicate with computers which used an IBM-PC data bus. The PC-bus speed is about 4 MHz. The data acquisition software was written in 8088 assembly codes. Thus, it is concluded that this piece of hardware is incompatible with the 386-based computer which employs a different bus architecture than the XT.

2.1.4.2/ Proposed 386-Based DART -- The possibilities of using the LAB MASTER card in a 386-computer were discussed with TECMAR. The advice given by TECMAR was: 1) The data acquisition software must be revised in order to adjust the counter speed of the card. This is necessary because the I/O card is originally designed to communicate at the PC-bus speed which is much slower than the AT-bus speed (standard 386-machines use AT-bus at speed of 8 MHz). It was suggested by TECMAR that the likely solution is to add a 'jump two bytes' instruction with each 'in' or 'out' instruction. However, this is open for investigation. 2) Since the LAB MASTER card is not intended for the 386-computer, possibly there are other problems which are presently unknown.

It was decided to install a new interface card with the same specifications as the LAB MASTER card. The primary specifications include: the number of I/O channels; the throughput rate and the bit resolutions. The A/D card which meets these requirements is the DT2821 data acquisition card manufactured by DATA TRANSLATION.

For software support, there are a number of commercial packages which control the board operations of the DT2821 card. The ATLAB package, suggested by DATA TRANSLATION, was selected for the DART project. The package provides subroutine libraries to control the required I/O functions through user-written programs (such as Fortran).

It is important to note that the DT2821 interface card is compatible only with a 386-machine which uses an IBM-AT data bus. There are other 386-machines which employ different I/O bus architecture. For example, the IBM-PS/2 machine uses micro-channel architecture which is not compatible with DT2821.

2.1.5/ System Durability

The 386-based DART is recommended to follow the same industrial configuration as the XT-based system. This includes: the entire computer system being provided in a rack mounted heavy gage steel enclosure and the disk drives (both the floppy and hard disk drives) being shock and vibration mounted. Also, the system has to be reliable and repairable under any circumstances.

2.2/ Generic Specification for 386-Based DART

A generic specification for a 386-based computer system for DART is given below:

design: industrial computer configuration

CPU: 80386 processor

speed: at least 20 MHz

coprocessor: 80387 math coprocessor (at least 20 MHz)

system memory: at least 12 Mb RAM on board

hard drive: at least 100 Mb

floppy drives: one 5-1/4" 360 Kb and one 3-1/2" 1.44 Mb

I/O port: at least 1 serial/parallel/mouse port

I/O bus speed: IBM-AT compatible

monitor: multisync with VGA card

keyboard: enhanced keyboard

printer: wide carriage with high CPS

plotter: HP compatible

operating system: at least DOS 3.0

software: MS-Fortran version 5.0, Grafmatic and Plotmatic

utility packages: Norton Utilities and 386-MATLAB

2.3/ Data Acquisition Sub-System for 386-Based DART

The A/D interface card selected for the 386-based DART is the DT2821 board manufactured by DATA TRANSLATION. The reasons for choosing this card are: 1) DT2821 is designed to be used with the IBM-AT bus architecture; 2) it possesses all the features of the LAB MASTER card; and, 3) the I/O capabilities of the board can be controlled by a software package called ATLAB.

A general description of the features for the DT2821 are:

1) A/D throughput rate is 50 KHz

2) 16 single-ended or 8 differential A/D input channels

3) 12 bits resolution

4) 0 to 10 V unipolar and +/- 10 V bipolar

5) programmable gains of 1, 2, 4 and 8

6) two D/A channels with 12 bits resolution

However, the most attractive feature of the DT2821 is the dual DMA channel architecture which supports continuous performance data collection [11]. Continuous performance is a data sampling method which provides gap-free transfers of large volumes of data from memory or disk or to memory or disk without any loss of samples. The DMA interface of the DT2821 is compatible with 16-bit data transfers and can be jumper-selected to use DMA channel 5, 6, or 7 of the computer. The DMA buffers can be located anywhere in the 16 Mb memory space of the 386-computer and can be up to 64 K words long. It is because of this particular feature, the analog tape recorder can be eliminated. By exploiting the DMA operation, analog radar data can be digitized directly into the DART computer. The ATLAB software has routines that can be called from Fortran to perform functions such as data acquisition and buffers management.

The ATLAB is a real-time software package [12] providing a high level language interface (such as MS-Fortran) to the DT2821 board; whereas, the data acquisition procedures for the XT-based DART were written in 8088 assembly codes. The ATLAB's software allows the user to control all the analog and digital I/O capabilities of the interface card through operation-specific subroutines. Also, ATLAB works with buffers as opposed to registers. This greatly simplifies the programming, use, and application of the board. Some of the routines written in the ATLAB include: analog input/output; digital input/output; continuous sampling of data; buffers management; files management; DT2821/system clocks control; configuration setup and error processing.

2.4/ System Configuration Design for 386-Based DART

A block diagram given in Figure 1 illustrates the configuration of the 386-based DART system. The hardware includes:

. Superblue 386 model 320V industrial computer made by Transduction LTD. in Mississauga.

. B386S/20 20 MHz microprocessor with 387-math coprocessor

. 16 Mb RAM and 150 Mb hard disk

. DT2821 A/D interface card installed

. one 5-1/4" 360 Kb and one 3-1/2" 1.44 Mb floppy drives

. ATI VGA Wonder Video card

. NEC Multisync 2A monitor with monitor mounting kit

. industrial enhanced keyboard

. FUJITSU model DL2600 printer

. HP model 7475A plotter

. LOGITECH HI-RI mouse

. Tektronic model 2245A 100 MHz oscilloscope.

2.5/ Summary

In this chapter, a 386-based computer has been proposed to replace the existing XT-based DART system. The new system increases effectiveness and efficiency of DART operation. The 386-based DART possesses greatly increased processing speed, larger system memory and much more hard disk capacity. Also, by utilizing the routines from ATLAB, DMA operation can be programmed to control the DT2821 data acquisition interface card. All of these improvements allow radar signals to be digitized directly into the computer memory, thus the analog tape recorder can be eliminated. Finally, the system configuration design has been presented for the 386-based DART.

3/ The 386-Based Software Algorithms

The 386-based DART signal processing software maintains the same logic as its XT-based package. Two packages, DART90 and DECK90 written in Fortran-77, are developed for the project. The former performs data acquisition and signal analysis of radar data and the latter outputs results in a grid mapping format. In general, the codes for both packages are designed to take advantage of the processing capabilities of the new computer. The data acquisition algorithm utilizes the ATLAB routines in order to facilitate the DT2821 A/D interfacing card. The main features, flow diagrams and operation procedures of each program module will be discussed in the following sections and the source codes are provided in the appendix.

3.1/ Main Program - DART90

The main program, DART90, is the signal processing package. The flow diagram is given in Figure 2. One important note is that the ATLAB allocates the full 15 Mb extended memory of the computer into 120 64 K word buffers for DMA operation. Since one complete radar trace consists of 512 16-bits samples, each data buffer can hold a maximum of 128 waveforms. With this limitation, the processing algorithms are designed to exploit the full capacity of the system by handling bridge decks between: i) at least 12 m to at most 1.5 Km in length per line if 10 cm fifth wheel pulses are engaged; ii) at least 120 m to at most 15 Km in length per line if 1 m fifth wheel pulses are engaged or iii) at least 1.2 Km to at most 150 Km in length per line if 10 m fifth wheel pulses are engaged.

Upon the initiation of DART90, the user is prompted with a number of inputs (Figure 3). These inputs are well defined and are described as following:

1. Title for bridge deck or site is required for reference purposes (Figure 3(a)).

2. User supplied incremental distance measurement (Figure 3(b)). The fifth wheel is equipped with three distance pulses: 10 cm, 1 m and 10 m.

3. User supplied thickness from asphalt core sample which is used to initiate the signal processing algorithm (Figure 3(c)).

4. The thickness from the concrete core sample is optional. The algorithm will automatically search for the rebar reflection, if no information is provided. However, if the thickness from concrete core sample is given, it will definitely benefit the processing algorithm (Figure 3(c)).

5. The maximum number of lines allowed per site is 20 (Figure 3(d)).

6. Two modes for obtaining data: directly sampling radar waveforms or using a stored data set. For the first case, radar data are sampled and stored in the system memory. Upon the completion of each line, the data are retrieved from the system, copied to a data file for record, and analysed. For the second case, data file names and their size (bytes) are supplied by the user (Figure 3(d)).

7. A debugging feature is included in the package. If this feature is selected, a detailed report (such as the location and amplitude of each peak, being selected by the algorithm, of each processed waveform) is stored in the file DEBUG.mto (Figure 3(d)).

8. The file which contains the antenna internal reflections data set is required for performing waveform correction (Figure 3(d)).

9. The flat plate amplitude can be inputted via three modes: i) by sampling the flat plate signal directly; ii) by reading a known data file which contains the

pertinent flat plate data; or iii) by giving the flat plate amplitude directly (Figure 3(e)).

Having responded to the above prompts, DART90 proceeds to invoke several program subroutines to perform the following tasks: i) subroutine FPLATE90 to measure flat plate amplitude; ii) subroutine SIGNAL90 to acquire radar signals and perform waveforms analysis; and iii) subroutine TABLE90 to report processed results. The functions of each of these subroutines are discussed next.

3.1.1/ Subroutine FPLATE90

The primary function of the sub-program FPLATE90 is to obtain the voltage of the surface reflection from the flat plate waveform. This measurement is important for calculating the dielectric constant of asphalt. Twenty waveforms are sampled and averaged to give the required flat plate amplitude. The analog signals are digitized by the DT2821 interfacing card through software control by the ATLAB routines.

The ATLAB routines initialize and reset the DT2821 board such that it has a sampling frequency of 25 kHz, and gain of one at the A/D channel 0. The radar data are acquired via burst mode DMA operation. Upon the completion of each DMA cycle, the digitized samples are transferred from user buffer to an extended memory buffer such that the user buffer can be released for the next DMA operation. After 20 waveforms have been sampled, data in the extended memory buffer are retrieved to the user buffer for data processing.

The radar data are known to contain antenna internal reflections which have to be removed prior to performing signal processing. The method proposed in [9] for suppressing antenna internal reflections is followed and is implemented in the subroutine REMIR90. Figure 4(a) illustrates the raw data superimposed with the antenna internal reflections waveform and Figure 4(b) shows the corrected waveform after subtracting the antenna internal reflections waveform from the radar data set. The flow graph for subroutines FPLATE90 is given in Figure 5.

3.1.2/ Subroutine REMIR90

The removal of antenna internal reflections from the radar signal is carried out in the subroutine REMIR90. A very simple first order subtraction is used in the process. The flow diagram is given in Figure 6. A pre-recorded antenna internal reflections signature is stored in the system. This data set is retrieved and subtracted from each radar waveform over a three-sample set: i) exact case - line up both waveforms with respect to the first sample; ii) left shift case - radar waveform shifted one sample to the left with respect to the first sample of the antenna internal reflections waveform and iii) right shift case - radar waveform shifted one sample to the right with respect to the first sample of the antenna internal reflections waveform. The root-mean-square (rms) values over the first 116 samples are computed and compared for the three cases. A corrected radar waveform is obtained by subtracting the entire data set where the rms value is a minimum from the above comparisons.

3.1.3/ Subroutine SIGNAL90

The primary functions of the subroutine SIGNAL90 are: i) to conduct data acquisition and ii) to perform signal analysis. The entire 15 Mb extended memory of the system is reserved for the purpose of data acquisition. There are 120 data buffers set up, by ATLAB, in the extended memory with each buffer having 64 K word. Four of the A/D channels of the DT2821 board are selected for inputting analog signals from the radar and fifth wheel systems to the computer. Channels 0, 1, 2 and 3 are dedicated for radar video, radar pre-trigger, fifth wheel pulse and line sync signal respectively. ATLAB routines are called to control all analog I/O functions of the DT2821 board. It is noted that since the system does not require any back-up device for data (analog) collection, the software is then designed to store digitized radar waveforms onto drive E (33 Mb capacity) of the computer.

The flow diagram for SIGNAL90 is given in Figure 7.

The steps for data acquisition are as follow:

. Initialize and reset the DT2821 board. Set sampling rate to 25 kHz and gain to one at all I/O channels.

. Declare one user buffer to size 512-word and 119 extended memory buffer each of size 64 K word.

. Input at Channel 3 of the DT2821 board is sampled continuously until the line sync signal is high.

. Input from the fifth wheel pulse at Channel 2 is sampled until the rising edge of the signal is detected.

. Input from the radar pre-trigger signal at Channel 1 is sampled until the rising edge of the signal is detected.

. Acquire 512 samples of radar signal at Channel 0 using DMA burst mode and store temporarily in the user buffer.

. Copy the data from the user buffer to the extended memory buffer upon the completion of one DMA cycle and the user buffer is released for the next DMA operation.

. Perform book-keeping for each extended memory buffer.

. Monitor input at Channel 3. If the line sync is low, then terminate the data acquisition process.

The steps for signal processing are as follow:

. Retrieve data from the extended memory buffer to the user buffer and copy the data to drive E.

. Reset all the parameters, flags and counters.

. Obtain the corrected radar waveform by subtracting the antenna internal reflections using the procedures described in the subroutine REMIR90.

. Invoke the subroutine PEAKCD90 to locate the surface reflection peak; asphalt-concrete interface peak and rebars reflection from every set of 10 consecutive waveforms. These parameters are averaged and the results are used as a guide for the second pass of the peaks searching algorithm.

. Employ the subroutine PEAKS90, using the average parameters, to conduct a fine tuning of peak searching from the same set of consecutive waveforms.

. Obtain the thickness of asphalt and concrete covers for every 1 m average and per line average.

. Evaluate the deck and identify faults, such as delamination, scaling and debonding, for each line.

The processed results are arranged into three groups and stored in three different data files, namely PLOT.mto, CONDIT.mto and TABLE.mto. The PLOT.mto file contains information related to number of lines; fifth wheel pulses and number of waveforms for each line. The CONDIT.mto file contains the identification, such as good, scaling, delamination and debonding, of each waveform being processed. These two files are used in the DECK90 package for generating a map layout of bridge deck condition. The third data file TABLE.mto, which is used by the subroutine TABLE90 for preparing summary report for the overall bridge deck assessment, stores the processed results including thickness measurements and deck conditons.

3.1.4/ Subroutine PEAKCD90

The surface reflection (peak A), transient peak due to surface reflection (peak B), asphalt-concrete interface reflection (peak C) and reflection from rebars (peak D) are located using subroutine PEAKCD90. The flow diagram is given in Figure 8. It has been found that peaks A and B are very easy to locate since they are relatively consistent from all previous observations [7-9]. However, this is not the case for peaks C and D. As suggested in [9], and followed here, asphalt and concrete core samples should be taken and measured on each bridge site in order to obtain a rough estimate of their thicknesses. These estimates serve as an initial searching point for both peaks C and D in the algorithm.

The sample locations of peaks C and D are averaged over a set of 10 waveforms. The average values are used to further identify the peak location defined in subroutine PEAKS90.

3.1.5/ Subroutine PEAKS90

Using the average sample locations of peaks C and D given by subroutine PEAKCD90, a fine tuning is conducted in identifying those reflections from each of the relevant waveforms. Other parameters, such as the ratio 'Rone' of magnitude at peak B to magnitude at peak A; the ratio 'Rtwo' of magnitude at peak C to magnitude at peak A; dielectric constant of asphalt; asphalt thickness; concrete cover over rebars; delamination; debonding and scaling are also evaluated. The flow diagram is given in Figure 9.

The selected threshold values for ratios 'Rone' and 'Rtwo' for declaring debonding are - 'Rone' is less than or equal to 0.32 and 'Rtwo' is greater than 0.15. For declaring scaling the selected threshold values are - 'Rone' is greater than 0.32 and 'Rtwo' is greater than 0.15. The presence of the 'W-characteristic' between peaks C and D is again used in discriminating good and delaminated concrete [3].

3.1.6/ Subroutine TABLE90

This module simply retrieves and re-organizes the processed results stored in the data file TABLE90.mto. The output provides a summary, in tabular format, of the overall conditions of the bridge deck. The summary is also stored in the file SUMMARY.mto. The flow diagram is given in Figure 10.

3.2/ Output Program - DECK90

The functions of DECK90 are i) to plot a bridge deck layout in grid mapping format and ii) to indicate the areas of deterioration using the results from DART90. The plotting routines utilize the PLOTMATIC graphic package. The flow diagram is given in Figure 11. Two data files, PLOT.mto and CONDIT.mto, are required by DECK90. The first file provides the number of lines to be plotted and the length of each line. The second file bears the symbols which represent the three states of bridge deck conditions: 'S' for scaling, 'B' for debonding and 'X' for delamination. These symbols are marked along each grid line according to the information provided by CONDIT.mto. The only input required from the user is the displacement of each line from the curb.

3.3/ Summary

In this chapter, the software algorithms for the 386-based DART were presented. The new software maintains the same logic as its XT-based package.

There is one fundamental improvement in the new 386-based software. Instead of using assembly codes as for the XT-based DART, the data acquisition procedures for the 386-based DART are developed using the ATLAB routines which are Fortran callable. Also, with the new configuration the software is designed to be more flexible with features such as: i) sampled radar data directly through DMA operation using the extended memory of the system; ii) copied the digitized data onto drive E of the computer for permanent record storage, thus eliminating the need for analog data recording; and iii) increased resolution by permitting waveforms to be sampled once every 10 cm.

4/ Processing of the Ninth Line Bridge Deck Data

Survey results for Ninth Line underpass (site 10-82-327) near Hwy 403/407 interchange in Oakville is provided in this chapter. The bridge deck has two lanes and is asphalt covered and waterproofed. It is expected that the deck should be in good condition since it was built in 1984. Data were collected and processed using the 386-based DART. The results are compared with the previous evaluation obtained by the XT-based system. This step is to ensure that for a known deck, the new system is functioning properly and operates satisfactorily in the field. In addition, the results are compared with the assessment from Penetradar's processing strategy.

4.1/ Survey Using the XT-Based DART

In the Spring of 1987, bridge deck assessment using the XT-based DART was conducted at Ninth Line. The survey was reported in [9]. Eight lines were collected, with line 1 at 1.4 m; line 2 at 2.6 m; line 3 at 3.3 m; line 4 at 4.9 m; line 5 at 6.2 m; line 6 at 7.4 m; line 7 at 8.9 m and line 8 at 10.05 m, respectively, from the east side barrier wall. Each line was found to be approximately 64 m in length.

The software designed for the XT-based DART sampled the waveforms once every 20 cm. Signal processing was done off-line, i.e. data were recorded on analog tapes and played back for data acquisition. A total of 320 radar waveforms were collected from each pass using the segmentation procedure (see Section 2.1.1) in order to cope with the limited capacity of the system memory. One complete pass of Ninth Line data required about 15 minutes processing time. Each waveform was processed: i) to measure the asphalt and concrete covers and ii) to identify the condition such as good, scaling, debonding or delamination. Both asphalt and concrete covers were averaged over every five waveforms to yield thickness measurement for each metre displacement on the deck.

The survey obtained from [9] is included here: i) Table 1 gave the survey summary of the bridge deck and ii) Figures 12 and 13 indicated areas which were identified delamination and debonding by the analysis. The computed average of asphalt thickness was 99 mm and concrete cover was 83 mm. The overall condition of the bridge deck was found: 97.24% good deck, 1.36% debonded deck and 1.28% delaminated deck.

Two sample waveforms from line 2 [9] are provided in Figure 14. Note that both waveforms possess a very distinct 'W-characteristic'. Consecutive waveforms from selective areas on lines 3 and 5 [9] are given in Figures 15 and 16. Each waveform (progressing from bottom to top) represents 20 cm displacement along the deck. These results are compared with the survey done using the new 386-based DART described in the following section.

4.2/ Survey Using the 386-Based DART

The 386-based system was field tested at Ninth Line on September 17th, 1990. With a van speed of 5 Km/hr, ten passes of data were collected at the site. The first pass was 1.2 m from the east side barrier wall and the remaining passes were 1 m apart. (i.e. line 2 is 2.2 m from the barrier wall or 1 m from line 1 etc.).

One major improvement in the new system is that a complete pass of radar data can be digitized and stored into the extended memory using the DMA method. After one pass of data collection is concluded, the sampled data residing in the extended memory are retrieved to the system memory and signal processing is performed immediately. It is noted that the segmentation procedure required by the XT-based DART has been eliminated. The digitized radar waveforms are copied into separate data files, on drive E of the computer, which are labelled according to their respective line numbers. (For example, data from line 1 and line 2 are stored in file line1.mto and line 2.mto, respectively).

The new software samples the waveforms once every 10 cm. A total of about 640 waveforms were collected from each pass of Ninth Line. The processing time for evaluating one complete pass of bridge deck data requires only 1.2 minutes which is about 25 times faster than the XT-system.

4.2.1/ Bridge Deck Profile

In this section, consecutive radar waveforms from each of the 10 passes collected from Ninth Line are examined. The left-hand-side of the set of plots from Figures 17 to 26 gives the stacking waveforms of individual lines. Progressing from the bottom to the top, each waveform represents 1 m displacements obtained from the averaging of 10 waveforms over each metre. These plots demonstrate a two-dimensional profile of the bridge structure. Thus, any variations in the asphalt and concrete covers can be easily seen. The right-hand-side of the set of plots gave the overall average waveform of each pass.

The antenna internal reflections of each waveform are removed using the procedures outlined in [9]. The surface reflection peaks of the sets of waveforms are aligned at the 5.4 ns time mark in order to provide a common reference point. The flat plate reflection at Ninth Line is found to be 7.93 V. From the average waveforms obtained from the profile plots, the surface reflection peak is measured to be 3.5 V. Thus, the dielectric constant of asphalt is calculated to be 6.66. It is noted that the asphalt-concrete interface peaks and the rebar reflections are located at about 7.25 ns and 8.75 ns, respectively, as seen in the profile plots. The dielectric constant of concrete is assumed to be 8.22 [9].

Several particulars are observed from the sets of plots: i) The waveforms shown on lines 1 to 4 and lines 8 to 10 have well defined 'W-characteristics' between the asphalt-concrete interface peaks and the rebar reflections; ii) The 'W-characteristic' feature between 25 and 40 m from lines 5 to 7 are less defined which might signify delamination at the vicinity; iii) Reflections, in some waveforms around the 40 m mark from lines 3, 5, 6 and 7, are quite prominent at the asphalt-concrete interface. This is usually a sign of debonding or scaling depending on the magnitude ratios measured by the subroutine PEAKS90; and iv) The interface between the waterproofing membrane and bottom of the asphalt might be deteriorated in some areas along lines 8 to 10 where variations in the reflections are seen at the interface.

Waveforms obtained from lines 3 (Figure 19) and 6 (Figure 21) can be used to compare with the data from lines 3 (Figure 15) and 5 (Figure 16) collected by the XT-based DART. These two sets of data covered the same areas on the bridge deck. The comparisons show a good agreement as seen by the presence of the strong 'W-characteristics' in both sets of waveforms from line 3 and the less defined rebar reflections from the waveforms in lines 6 (from 386-based) and 5 (from XT-based). It is noted that any significant change in the radar signatures found in two separate surveys (usually a few years apart) should require attention as this might indicate an onset of deterioration.

4.2.2/ Waveforms Analysis

4.2.2.1/ Thickness Measurements -- The major parameters related to the calculation for asphalt and concrete covers are: i) The surface reflection peak (peak A); ii) The time delays of the asphalt-concrete interface reflection (peak C) and rebar reflection (peak D) with respect to the surface reflection; and iii) The dielectric constant of asphalt and concrete.

The procedures for calculating the dielectric constant of asphalt, given in [9], are implemented in the software. Since the radar signal strength attenuates as it penetrates and propagates through the asphalt, the dielectric constant of concrete is very difficult to measure. It was reported in [9] that the dielectric constant of concrete was estimated to be 8.22.

Figures 27 to 31 plot the locations of peaks A, B, C and D of every waveform for each of the 10 passes of data. The calculated asphalt and concrete covers are also provided at the bottom half of the plots. It is noted that both peaks A and B are very consistent. The variations at peaks C and D can be reduced by employing the searching algorithm discussed in section 3.1.4. From the profile plots, given in the previous section, the asphalt cover is estimated to be 100 mm and concrete cover is about 85 mm.

4.2.2.2/ Faults Identification -- Faults due to scaling and debonding are commonly found in the vicinity of the asphalt-concrete interface. In previous studies [3, 5], it is suggested that the magnitude at peaks A, B and C from the radar waveforms are directly related to the presence of scaled or debonded sub-surface. Identification of these two types of deterioration can be determined by calculating the two ratios - 'Rone' and 'Rtwo': where 'Rone' is the ratio of magnitude at peak B to magnitude at peak A and 'Rtwo' is the ratio of magnitude at peak C to magnitude at peak A.

Figures 32 to 36 are the 'Rone' and 'Rtwo' ratios for each of the 10 passes from Ninth Line. It is noted that the norms for 'Rone' ranges from 0.25 to 0.32 and for 'Rtwo' ranges from 0.0 to 0.15. The bridge deck was expected in fairly good condition, since it was built in 1984. Thus, the norms should reflect good asphalt-concrete interface. Any values beyond the norms may indicate abnormalities at the locality. It has been verified in [5] that relatively high values in 'Rone' and 'Rtwo' may be caused by a scaled concrete surface. A relatively high value in 'Rtwo' alone may be due to a debonded interface. Therefore, by selecting a pair of threshold values for the two ratios, scaled and debonded sub-surfaces can be identified.

Originally, the threshold values for 'Rone' and 'Rtwo' assigned for the XT-based DART were 0.40 and 0.28 respectively. Since a completely different configuration is used in the new system, the threshold values may deviate from the XT-based system. In the subroutine PEAKS90, the threshold values for 'Rone' and 'Rtwo' are selected empirically: Scaled sub-surface is where 'Rone' is greater than 0.32 and 'Rtwo' is greater than 0.15; and, Debonded interface is where 'Rone' does not exceed 0.32 but 'Rtwo' is greater than 0.15. The threshold levels are also indicated on the plots. From the ratios test, a combination of scaled and debonded sub-surfaces are identified from the data set. A majority of these deteriorations are found on lines 3, 5, 6 and 7.

The 'W-characteristic' between peaks C and D is associated with sound concrete at the sub-surface. The presence of delamination defeats the 'W-characteristic' [5]. Thus, faults due to delaminated areas can be detected simply by identifying the absence of the 'W-characteristic'. It is noted that the 'W-characteristic' is commonly found in asphalt-covered concrete bridge decks with asphalt thickness in the range of 50 to 150 mm and concrete cover over reinforcement of the order of 60 mm.

4.2.3/ Bridge Deck Assessment

Data collected from Ninth Line were processed using the DART90 algorithm described in chapter three. From the profile plots given in section 4.2.1, the asphalt and concrete covers were estimated to be 100 and 85 mm, respectively. Both of these values were substituted for the core sample measurements requested by the software. The 386-based DART provides much better resolution than its predecessor in measuring both asphalt and concrete covers since instead of computing the average measurement over every five waveforms as being done in the XT-system, the 386-software performs a 10-waveform averaging.

Table 2 provided the survey summary. It is observed that: 1) for wheel rut lines (lines 3, 5, 7 and 9), the asphalt is thinner than for the other lines; 2) the asphalt between the wheel ruts (lines 4, 6, and 8) is thicker than that of the wheel rut lines; 3) the asphalt on the shoulder (lines 1, 2 and 10) is the thickest of all lines.

The measured concrete cover obtained from Table 2 is seen to be fairly consistent except in some parts of lines 4, 5 and 6 where the rebar reflections cannot be determined easily (as shown by the profile plots).

In comparing the results obtained from Table 2 with those from Table 1 (XT-based DART), it is noted that there is little difference in the asphalt thickness estimates. For the concrete cover, it is believed that the 386-system yields better results than the XT-system. This is due to the fact that a core sample measurement from concrete cover was not included in the XT-system at that time.

The overall average asphalt thickness is 97 mm and concrete cover is 78 mm. Delamination is 3.80% and is found mostly along lines 5 to 8. Debonding is 3.53% and is found mostly along lines 3, 5 and 6. Scaling is 2.74% and is also found along lines 3, 5 and 6. The bridge deck is 89.93% in good condition. In comparing with the previous survey (Table 1) taken in 1987 there is a 7% drop in the overall condition of the bridge deck indicating that further deterioration has since taken place.

The areas where faults were detected by the algorithms are indicated in Figures 37 to 39. It is noted that both scaled (indicated as 'S' in Figure 37) and debonded (indicated as 'B' in Figure 38) areas are clustered around the same localities. The scaled areas may actually be debonded areas where trapped moisture gives radar signatures that resemble scaling. The radar waveform was observed to have greatly increased magnitude for both reflections at peaks B and

C. Consequently, the ratios 'Rone' and 'Rtwo' exceed the threshold level which is assigned for detecting scaling. The debonded areas shown on lines 3 and 6 are also found in the survey obtained by the XT-system (Figure 13). Faults due to delamination (indicated as 'X' in Figure 39) are particularly active, in comparing with the previous survey (Figure 12), around the centre section of the south bound lane.

Four areas from each of lines 4, 5 and 6 are selected such that their radar signatures can be examined in order to justify the processed results. These areas cover good, delamination, scaling and debonding as marked in Figures 37 to 39. Waveforms from 12, 18, 30 and 40 m on line 4 are plotted in Figure 40. The well defined 'W-characteristic' between peaks C and D indicates that the areas are in good condition. The same locations are chosen for line 5 in which Figure 41 shows the waveform at 12 m (delamination), 18 m (debonding), 30 m (delamination) and 40 m (scaling). Waveforms from line 6, shown in Figure 42, are at 25 m (delamination), 35 m (delamination), 39 m (scaling) and 45 m (debonding). It is noted that waveforms from scaled areas are characterized by the prominent reflections both at peaks B and C; and, whereas waveforms from debonded areas also have prominent reflections at peak C and, to a lesser extent, at peak B. Waveforms from delaminated areas are distinguished by the absence of the 'W-characteristic'.

4.3/ Comparison with Penetradar's Results

Penetradar conducted a survey to the Ninth Line bridge deck June 14th, 1990. The radar inspection report is provided in Table 3. Twelve passes of data were collected and analyzed for that survey. The results from Penetradar and DART are compared. Passes 1 (line A) and 12 (line L) in Penetradar's survey are not covered by DART. Therefore, comparison starts at pass 2 (line B) which is line 1 in DART and ends at pass 11 (line K) which is line 10 in DART. For simplicity, lines 1 to 10 are used as the convention. Furthermore, it is noted that DART presents waveforms once every 10 cm to its processing algorithm; whereas Penetradar presents waveforms once every 50 cm. Therefore, DART provides more coverage in area and, statistically, it yields a greater minimum estimate on the condition of the bridge deck. On a badly deteriorated bridge deck, DART would provide a larger percentage of deterioration in comparing with Penetradar's survey.

From Penetradar's result, the overall condition of the bridge deck from line 1 to 10 is 87.35% good; 3.25% delamination; 9.40% debonding and no scaling. The overall average from line 1 to 10 for asphalt thickness is 90 mm and concrete cover is 80 mm. The delaminated and debonded areas are shown in Figures 43 and 44, respectively.

The overall average asphalt thickness computed by both systems is within +/- 3.61% difference. The least variation among the two measurements is found on line 9 where +/- 1.10% difference is recorded. The maximum variation is found on line 8 where +/- 6.77% difference is registered. For concrete cover, the overall average is only +/- 1.28% difference. Both systems yield the same measurement on line 3. The maximum variation of +/- 9.09% is found on line 7.

DART indicated delamination on every line with the most deterioration on lines 5, 6, 7, and 8. From the profile plots given in section 4.2.1, it is noted that the 'W-characteristic' is badly defined along these lines. The survey identified a total of 3.80% delaminated deck by area; whereas Penetradar detected 3.20% delamination scattering along lines 2, 3, 4, 8, 9 and 10. Nevertheless, both systems indicated that delamination was found mostly on the south bound lane.

The areas where Penetradar claimed debonding were also identified by DART. This was especially true along lines 3, 5 and 6. It is noted that the scaled areas detected by DART could actually be due to the accumulation of moisture in the debonded areas. If the results were combined for scaling and debonding, then DART identified a total of 6.27% debonding; and, whereas Penetradar gave 9.40% debonding.

4.4/ Summary

In this chapter, the survey from the Ninth Line bridge deck using the 386-based DART was presented and compared with the previous survey done by the XT-system. It was noted that the 386-based DART out-performed its predecessor and operates effectively and efficiently in the field for a known deck. In addition, the survey was also compared with the results obtained from Penetradar's radar system. It was found that both systems yielded similar performance in measuring both asphalt and concrete covers and identifying faults.

5/ Processing of the Bobcaygeon Bridge Decks Data

Two bridge decks at Bobcaygeon (north of Lindsay) were surveyed on August 30th, 1990. The bridge decks, commonly known as the Little Bob South bridge (site 32-079) and the Little Bob North bridge (site 32-080), were built in the 1950s. Both bridges, originally exposed concrete, are now asphalt covered and waterproofed. Ten passes of data were collected from each site. Each pass was 1 m apart starting from the east side barrier wall. The total length of each deck is about 24 m. There were no cores taken on both sites. However, a number of core measurements were obtained from the Ministry's record showing that asphalt cores of 90, 120, 100 and 90 mm were obtained from the Little Bob South bridge and 100 and 200 mm were obtained from the Little Bob North bridge. The actual locations of where the cores were taken was not provided.

The assessments for both bridge decks given by the 386-based DART are presented in this chapter. In addition, the results are compared with the surveys evaluated by Penetradar who performed their radar inspection of the two bridges on the same day as DART.

5.1/ Assessment of the Little Bob South Bridge

Data collected from the Little Bob South bridge were first examined using the profile plots. The cores sample measurements were substituted by the estimations obtained from the profile plots. Results from the radar inspection are provided, discussed and compared with Penetradar's survey. Sampled waveforms representing different conditions on the bridge deck are also examined.

5.1.1/ Bridge Profile Plots

Waveforms from each of the 10 passes are given in Figures 45 to 54. There are about 240 waveforms collected from each pass of data. The plots follow the format given in section 4.2.1. The surface reflection peaks and asphalt-concrete interface peaks can be identified without much difficulty. However, it is noted that the rebar reflections from some of the waveforms are difficult to detect.

The flat plate signal was found to be 9.38 V and the average amplitude of the surface reflection peak was about 3.75 V. Thus, the calculated average dielectric constant of asphalt is 5.44 and the assumed value for the dielectric constant of concrete is 8.22. The average location of the asphalt-concrete interface peaks is shown to be at 7.25 ns. Therefore, the asphalt cover is estimated to be approximately 120 mm.

The profile plots also reveal that the depth of the rebars varies from one end of the bridge deck to the other end. They tend to follow a curved pattern where the rebars for each pass are deeper in the first and the last 5 m. The rebar reflections near the two ends of the bridge deck are seen mostly around the 9 ns region. On the other hand, the reflections from the centre section of the bridge deck are relatively constant at 8.25 ns. Figure 55 exhibits a detailed view of the curved pattern from lines 3 and 4 where each trace is the average waveform from every 50 cm. The concrete cover is estimated to range from 53 mm at the centre section to 117 mm near the two ends of the bridge deck.

5.1.2/ Radar Inspection

The computer survey for the Little Bob South bridge is given in Table 4. The core sample inputs required by the algorithms were substituted by the measurements obtained from the profile plots where the asphalt thickness was found to be 120 mm and concrete cover was found to be 60 mm. From Table 4, it is noted that the computed asphalt thickness agrees very well with the observations from the profile plots. The computed values for the concrete cover demonstrate a curved pattern where the measurements are about 90 to 100 mm at both ends of the bridge deck and 45 to 55 mm at the centre section. In this case, attention should be given in interpreting the average concrete cover measurement. The overall result indicates 47.17% good deck by area; 21.15% scaled deck by area; 28.90% delaminated deck by area and 2.77% debonded deck by area. The average asphalt thickness is 119 m and concrete cover is 71 mm. It is found that lines 1, 9 and 10 have thicker concrete cover than the other lines.

The areas identified by the algorithms as scaling, debonding and delamination are given in Figure 56 to 58. It is shown that scaling is very dense at the south end of the bridge deck. Debonding is not acute and is around the area where scaling is detected. A high percentage of delamination is detected around the centre section of the bridge deck. Sampled waveforms from good, delaminated, debonded and scaled areas are given in Figures 59 to 62, respectively.

5.1.3/ Comparison with Penetradar's Results

Penetradar's results for the Little Bob South bridge are given in Table 5 and Figure 63. The average asphalt and concrete cover reported by Penetradar are 88 and 76 mm, respectively. Penetradar indicates that the bridge deck is 21.46% scaled deck by area; 18.96% delaminated deck by area and no debonding.

In comparing the results between Penetradar and DART, for the asphalt cover: there is +/- 13.03% difference in measuring the overall thickness; the least variation is found on line 9 where there is only +/- 9.82% difference; the maximum variation is found on line 10 where there is +/-19.13% difference; and, for the concrete cover: there is +/- 3.52% difference in measuring the overall thickness; line 7 from both surveys yielded similar line average; the maximum variation of +/- 15.67% is found on line 2.

It is believed that the discrepancy between the two results in measuring the asphalt thickness is mainly due to the value used for the dielectric constant of asphalt. DART computes the dielectric constant of asphalt for each waveform;

whereas Penetradar calculates the dielectric constant of asphalt once, and bases its calculation on the measurements obtained by taking core samples. Furthermore, waveform correction (i.e. subtracting the antenna internal reflections from each waveform) is not performed by Penetradar. The actual value for the dielectric constant of asphalt used by Penetradar is not available. For the concrete cover, both surveys provide good agreement with each other. It is noted that Penetradar and DART use the same value for the dielectric constant of concrete.

Both Penetradar and DART are in good agreement in locating scaled areas. The debonded areas found by DART are located in the vicinity of scaled areas. From experience, it is less likely to detect debonding of bridge deck over 35 years old; whereas scaling can be significant. It is believed that those debonded areas could actually be scaled areas. The radar waveforms from these areas were shown to have very prominent reflection at peak C and less prominent reflection at peak B. DART uses two ratios, 'Rone' and 'Rtwo', to discriminate scaling and debonding; whereas Penetradar only measures the change in magnitude at peak C for determining scaling and debonding. Obviously, DART is very sensitive, in comparison to Penetradar, in identifying scaling and debonding.

DART detected 10% more delamination than Penetradar. This is to be expected due to a finer resolution (i.e. 10 cm vs 50 cm) by DART, who found delamination was found very active in the centre section of the bridge deck. From the plots given in section 5.1.1, it is observed that waveforms from these areas show the absence of the 'W-characteristic' feature. The areas where reported delamination by Penetradar were also claimed by DART.

5.2/ Assessment of the Little Bob North Bridge

In this section, the survey from the Little Bob North bridge is presented and the results are compared with Penetradar's radar system.

5.2.1/ Results from 386-Based DART

Figures 64 to 73 provide the profile plots for the Little Bob North bridge. The flat plate signal was found to be 9.68 V and the average amplitude of the surface reflection peak was about 3.80 V. The calculated average dielectric constant of asphalt is 5.24 and the assumed value for the dielectric constant of concrete is 8.22. From the profile plots, the average location of the asphalt-concrete interface peak is located at around 7.15 ns. Therefore, the estimated asphalt cover is 115 mm. A similar curved pattern as seen in the South bridge's data for the depth of the rebars is also observed here. The rebar reflections are found to locate from 8.5 to 9.5 ns near the two ends and at 8.25 ns near the centre section of the bridge deck. The concrete cover is estimated to vary from 57 to 122 mm.

The results from the radar inspection are included in Table 6. The core sample measurements were substituted by the estimated values obtained from the profile plots where the asphalt thickness is estimated to be 115 mm and the concrete cover is estimated to be 60 mm, respectively. The overall condition of the bridge deck is 54.58% good deck by area; 25.51% scaled deck by area; 5.68% debonded deck by area and 14.40% delaminated deck by area. The computed average asphalt thickness is 115 m and concrete cover is 82 mm. Again, attention should be given to the average value of the concrete cover as the rebars are deeper near the both ends and shallower near the centre section of the bridge deck.

Areas identifying scaling, delamination and debonding by DART are shown in Figures 74 to 76. It is observed that scaling is very active near both ends of the structure. Debonding is found mainly on line 4. Delamination is seen around the centre section of the bridge deck. Sampled waveforms from the data set, representing good, delamination, debonding and scaling, are given in Figures 77 to 80.

5.2.2/ Comparison with Penetradar's Results

Penetradar's results for the Little Bob North bridge are given in Table 7 and Figure 81. The overall average asphalt and concrete covers reported by Penetradar are 93 and 89 mm, respectively. Penetradar indicates that the deck is 26.04% scaled by area; 20.00% delaminated by area and has no debonding.

In comparing the results between Penetradar and DART, for the overall asphalt cover: there is +/- 9.57% difference in measuring the thickness; the least variation among the two measurements is found on line 1 where there is only +/- 4.47% difference; the maximum variation is found on line 6 where there is +/- 16.95% difference; and, for the overall concrete cover: there is +/- 4.27% difference in measurement; the least variation of +/- 0.59% is found on line 7; the maximum variation of +/- 18.24% is found on line 9.

The discrepancy between the two results in measuring the asphalt thickness has been discussed in section 5.1.3. It is believed that the differences are due to the procedures applied in computing the dielectric constant of asphalt. Both system employs the same dielectric constant of concrete, which is 8.22, in computing the concrete cover; and, the results reflects a good agreement in the measurement.

There is +/- 5% difference in the amount of scaling identified by DART and Penetradar. However, both systems agree quite well in locating the scaled areas. Judging from the age of the bridge deck, it is believed that the debonded areas found by DART may be scaling. The two ratios, 'Rone' and 'Rtwo', used by DART is noted to be very sensitive in identifying scaling and debonding. Penetradar only depends on the increase in magnitude at peak C for determining scaling and debonding.

The order in which the processing algorithms for DART checks for deterioration is: scaling, delamination, good area and debonding. The technique used by DART to discriminate scaling from debonding is based on the 'Rone' and 'Rtwo' ratios discussed in Chapter 3. The ratio test strongly depends on the magnitude of the radar reflections and the assigned threshold levels. On many occasions, the degree of moisture at the damaged area affects the magnitude of the reflected signal and hence waveform interpretation. For an older bridge deck, the software is more likely to find deterioration due to scaling than debonding. Penetradar also measures the magnitude of the radar reflection at concrete surface as the indication for sign of scaling and debonding. In addition, they also include the age factor in making decision for the detection scheme.

Penetradar reported that delamination is confined to the shoulder lanes. DART not only found delamination on the should lanes, it also found delamination across the centre section of the bridge deck. Waveforms from those areas are shown to exhibit the absence of the 'W-characteristic' feature.

5.3/ Summary

In this chapter, two bridge decks were used to field test the effectiveness of the 386-based DART. It was found that the system operates satisfactorily in the field.

The results from DART and Penetradar demonstrated comparable performance in terms of measuring the overall asphalt thickness; measuring the overall concrete covers and identifying faults. It is believed that the differences in measuring the asphalt thickness are caused by the procedures in obtaining the dielectric constant of asphalt. DART computes the dielectric constant of asphalt for each corrected radar waveform; whereas Penetradar calculates the dielectric constant of asphalt once using core sample information. Furthermore, Penetradar does not include waveform correction in their processing strategy.

Both DART and Penetradar provided good agreement in measuring the concrete cover for the two bridge decks surveyed. It was noted both systems use the same value for the dielectric constant of concrete.

The results show that DART and Penetradar provided good agreement in locating scaled areas. In addition, DART also identified a small area of debonding. The ratio tests, 'Rone' and 'Rtwo', used by DART are found to be very sensitive than Penetradar in determining scaling and debonding. It was noted that Penetradar not only measures the change in magnitude at peak C for classifying scaling and debonding, it also included the age of the bridge deck in the detection scheme.

The 'W-characteristic' feature is used by the both systems in discriminating sound and delaminated concrete. DART found the centre sections of the two bridge decks were quite delaminated. Waveforms from that region exhibited an absence of the 'W-characteristic' feature.

6/ Conclusions and Recommendations for Further Research

6.1/ Conclusions

It is concluded that:

1. The XT-based DART system requires replacement in order to meet the present requirements of bridge deck assessment.

2. It is found that a 386-based computer system ameliorates the shortcomings of the XT-based DART.

3. The criteria for selecting the 386-based computer which meet DART's requirements have been identified. The criteria include: system memory; system speed; storage devices; data acquisition sub-systems and system durability.

4. The minimum system requirements for the 386-based DART have been examined and found to be: at least 12 Mb system memory; at least 20 MHz processor speed; at least 100 Mb hard drive for storage and; a new data acquisition sub-system which is compatible with an IBM-AT bus speed.

5. The data acquisition sub-system has been investigated in detail. It was decided that a DT2821 data acquisition card is the appropriate choice.

6. The data acquisition procedures can be simplified and yet made more effective by employing the ATLAB's routines which are Fortran callable. Thus, it is unnecessary to develop assembly codes to do the task.

7. The system configuration design for a 386-based DART is provided. The hardware is composed of: an industrial design 386-computer with 20 MHz CPU; a

16 Mb system memory; a 150 Mb hard drive; one 5-1/4" 360 Kb floppy drive and one 3-1/2" 1.44 Mb floppy drive; one DT2821 A/D interface card; one HP 7475A plotter and one Fujitsu DL2600 printer.

8. With increased processing speed, computer memory size and hard disk capacity, data acquisition can be made directly by computer using the DT2821 interfacing card and ATLAB's routines. Thus, tape recording analog data can be eliminated.

9. The XT-based signal processing software package has been revised to facilitate the capabilities of the new system configuration. The main features, operational procedures, and logic flow-charts of each program module for the 386-based software are outlined.

10. The ATLAB allocates the full 15 Mb extended memory into 120 64 K word buffers for DMA operation. Since one complete trace of the radar waveform is 512 samples, each data buffer can hold a maximum of 128 waveforms. With this format, the new algorithms are designed to exploit the full capacity of the system and are capable of handling bridge decks of: i) 12 m to 1.5 Km in length per line if 10 cm fifth wheel pulses are engaged; ii) 120 m to 15 Km in length per line if 1 m fifth wheel pulses are engaged or iii) 1.2 Km to 150 Km in length per line if 10 m fifth wheel pulses are engaged. The digitized data are stored temporarily in the extended memory and are retrieved to the system memory, upon the completion of data collection, for waveform analysis. The sampled waveforms are also copied to a set of data files which reside on drive E of the computer for archival purposes.

11. The 386-based DART allows waveforms to be sampled once every 10 cm instead of every 20 cm which was a limitation of the XT-system. Thus, better resolution can be obtained.

12. The threshold levels for ratios 'Rone' and 'Rtwo', in determining scaling and debonding, have been adjusted to 0.32 and 0.15, respectively, in the new software. The 'W-characteristic' is retained for discriminating radar waveforms from delaminated and good concrete.

13. The increased computer throughput permits on-site evaluation of bridge decks with the condition summary analysis being provided on a line-by-line basis.

14. Using the new system, data are collected and processed from the Ninth Line bridge deck in Oakville; Little Bob South bridge and North bridge in Bobcaygeon.

15. Comparing the evaluations of the Ninth Line bridge deck obtained by the XT-system from a previous survey and the 386-system, the upgraded system is 25 times faster in outputting results than its predecessor.

16. For the Ninth Line bridge deck, the 386-based DART measured the overall average asphalt thickness as 97 mm and concrete cover as 78 mm. Delamination is 3.80% and is found mostly along lines 5 to 8. Debonding is 3.53% and is found mostly along lines 3, 5, and 6. Scaling is 2.74% and is also found along lines 3, 5, and 6. The scaled and debonded areas are seen to cluster around the same localities. It is believed that the scaled area may actually be a debonded area where trapped moisture gives radar signatures that resemble scaling.

17. Penetradar found that the overall asphalt thickness and concrete cover, for the Ninth Line bridge deck, to be 90 and 80 mm, respectively. In comparing the survey between Penetradar and DART, the overall asphalt thickness is within +/- 3.61% difference and the overall concrete cover is within +/- 1.28% difference. For fault identification, DART found delamination on every line with lines 5, 6, 7, and 8 having the most damage. Waveforms from these areas exhibited a badly defined 'W-characteristic' feature. Penetradar yielded 3.2% delamination scattering

along lines 2, 3, 4, 9 and 10. Both systems indicated that delamination was found mostly on the south bound lane. The areas where Penetradar claimed debonding (9.40%) were also identified by DART. This was especially consistent along lines 3, 5 and 6.

18. The results from Bobcaygeon were compared with Penetradar's survey. For the Little Bob South bridge, DART found the overall average asphalt and concrete covers to be 119 and 71 mm, respectively. The deterioration includes: 21.15% scaling; 2.77% debonding and 28.90% delamination. The survey from Penetradar showed the overall average asphalt and concrete cover as 88 mm (which is +/- 13.03% different from the DART's result) and 76 mm (which is +/- 3.52% different from the DART's result), respectively; and, the deterioration includes: 21.46% scaling and 18.96% delamination.

19. For the Little Bob North bridge, DART found the overall average asphalt and concrete covers to be 115 and 82 mm, respectively. The deterioration includes: 25.51% scaling; 5.68% debonding and 14.40% delamination. The survey from Penetradar showed the overall average asphalt and concrete cover as 93 mm (which is +/- 9.57% different from the DART's result) and 89 mm (which is +/- 4.27% different from the DART's result), respectively; and, the deterioration includes: 26.04% scaling and 20.00% delamination.

20. It is believed that the discrepancy in measuring the asphalt thickness between DART and Penetradar is mainly due to the procedures in computing the dielectric constant of asphalt. For DART, the antenna internal reflection is subtracted from each waveform and the dielectric constant of asphalt is calculated. For Penetradar, the radar waveform is not corrected for the antenna internal reflections and the dielectric constant of asphalt is calculated once using the core sample information.

21. DART and Penetradar use the same value for the dielectric constant of concrete in computing the concrete cover. It was shown that both systems yielded good agreement in this measurement.

22. In conmparison to Penetradar, DART is very sensitive in determining scaling and debonding. DART measures the ratios 'Rone' and 'Rtwo' for discriminating scaling and debonding; whereas, Penetradar only measures the change in magnitude at peak [illegible] for identifying scaling and debonding.

23. It was found that Penetradar not only measures the magnitude of the radar reflection at concrete surface to identify scaling and debonding, but it also includes the age of the bridge deck to enhance its detection scheme. For an older bridge deck, scaling is known to be more probable than debonding; and, the reverse is applied to a fairly new bridge deck.

24. DART samples waveforms once every 10 cm and Penetradar samples wavefroms once every 50 cm. Therefore, DART provides more coverage in area and, statiscally, it yields a greater minimum estimate on the condition of the bridge deck. Thus, on a badly deteriorated deck, DART would give a larger percentage of deterioration in comparison to Penetradar, as was observed in the above comparisons.

6.2/ Recommendations for Further Research

It is recommended that:

1. A menu-driven software package be developed to incorporate the existing data acquisition and signal processing algorithms with a number of extra selections of functions which enhance DART's operation, such as: i) to check for the necessity of recalibrating the radar system; ii) to offer visual examination of sampled

waveforms for sub-surface identification purposes; iii) to produce bridge deck profile plots for asphalt and concrete cover recognition purposes which can be used, in conjunction with selection (ii), to replace cores sampling on bridge deck; and iv) to provide a preference for on-site evaluation (i.e. perform signal processing after each pass of data being collected) or off-site evaluation (i.e. save digitized data after each pass and perform processing later).

2. The possibility of employing mass storage sub-systems, such as cartridge tape drive, be investigated for archival of storage large bridge deck data sets. Thus, radar signatures from the past surveys can be compared and correlated with present surveys in order to provide a history of the bridge's structure.

3. Pattern recognition schemes be studied to facilitate the bridge deck profile plots and signal analysis. This can be associated with the strata plots (also known as the dot plots) a technique developed at the Communications Research Laboratory.

4. Data classification techniques be developed to enhance the present signal processing algorithms.

5. Application of neural networks to identify bridge deck structures [10] be further investigated for providing a front-end processing strategy for a more reliable DART system.

7/ References

1. Manning, D.G. and Holt, F.B., "Detecting Delamination in Concrete Bridge Decks," Concrete International, Nov. 1980, pp. 34-41.

2. Manning, D.G. and Holt, F.B., "Detecting Deterioration in Asphalt Covered Bridge Decks," Transportation Research Record, No. 899, 1983, pp. 10-20.

3. Chung, T., Carter, C.R., Manning, D.G. and Holt, F.B., "Signature Analysis of Radar Waveforms Taken on Asphalt Covered Bridge Decks," Ministry of Transportation, Report ME-84-01, Downsview, Ontario, 1984.

4. Carter, C.R., Amorosi, R. and Chung, T., "Calibration of Computer Program for Radar Waveform Analysis," Ministry of Transportation, Report No. ME-86-03, Downsview, Ontario, 1986.

5. Carter, C.R., Chung, T., Holt, F.B. and Manning, D.G., "An Automated Signal Processing System for the Signature Analysis of Radar Waveforms from Bridge Decks," Canadian Electrical Engineering Journal, vol. 2, no. 3, 1986, pp. 128-137.

6. Carter, C.R., and Chung, T., "Applications of Radar in Highway Design, Construction and Maintenance," Ministry of Transportation, Report No. ME-87-22, Downsview, Ontario, 1988.

7. Carter, C.R., Chung, T., Masliwec, A. and Manning, D.G., "Analysis of Radar Reflections from Asphalt Covered Bridge Deck Structures," Workshop on Ground Penetrating Radar, Ottawa, Ontario, May 24-26, 1988.

8. Chung, T. and Carter, C.R., "Analysis of Impulse Radar Reflections from Bridge Decks," Communications Research Laboratory Radar Workshop, McMaster

University, Hamilton, Ontario, October 12-13, 1988.

9. Chung, T. and Carter, C.R., "Radar Signal Enhancement for DART," Ministry of Transportation, Report No. ME-89-05, Downsview, Ontario, 1989.

10. Vrckovnik, G., Chung, T. and Carter, C.R., "An Application of Neural Networks to Impulse Radar Waveforms from Asphalt-Covered Bridge Decks," Proceedings of the International Joint Conference on Neural Networks (IJCNN), Vol. II, pp. 453-456, Washington D.C., January 15-19, 1990.

11. User Manual for DT2821 Series, Document UM-05073-G, Data Translation, Inc., Ninth ed., October 1989.

12. ATLAB User Manual SP0143 V01.10, Document UM-06028-B, Data Translation, Inc., Third ed., October 1989.

Table 1: Survey Summary for Ninth Line Evaluated by XT-Based DART (1988).

```
Ninth Line (Oakville): line 1 to 8 (Feb 18, 1988)
AVErage THICKness SUMMARY

        [ AVErage  THICKness (mm) ]
                  line number

metre :   1          2         3         4          5          6          7          8
......................................................................................
  1   : 109 ( 66)   94 ( 64)   87 ( 62)   85 ( 38)   92 ( 62)   87 ( 47)   92 ( 58)   90 ( 58)
  2   : 127 ( 83)  122 ( 73)  106 ( 69)  110 ( 90)  110 ( 99)  116 ( 82)  108 ( 71)  106 ( 79)
  3   : 134 ( 80)  119 ( 74)  112 ( 72)  116 ( 90)  109 (104)   91 ( 99)  109 ( 71)  109 ( 78)
  4   : 121 ( 88)  119 ( 75)  114 ( 71)  113 ( 94)  106 (104)  112 ( 72)  111 ( 73)  111 ( 75)
  5   : 120 ( 86)  108 ( 81)  110 ( 77)  115 ( 86)  111 (104)  111 ( 91)  113 ( 71)  110 ( 74)
  6   : 116 ( 81)  102 ( 80)  107 ( 73)  111 ( 87)  110 (103)  115 ( 86)  111 ( 71)  113 ( 72)
  7   : 114 ( 82)   97 ( 83)  101 ( 77)  110 ( 90)  103 (106)  116 ( 82)  111 ( 69)  110 ( 70)
  8   : 113 ( 93)  101 ( 80)  103 ( 70)  110 ( 90)  102 (105)  112 ( 80)  111 ( 67)  108 ( 66)
  9   : 113 ( 91)   96 ( 80)  104 ( 69)  108 ( 88)  101 (108)  109 ( 82)  111 ( 63)  110 ( 68)
 10   : 114 ( 90)   97 ( 82)  104 ( 68)  105 ( 90)  103 (107)  108 ( 82)  108 ( 64)  112 ( 66)
 11   : 111 ( 80)  100 ( 73)  105 ( 66)  106 ( 94)  106 (105)  107 ( 78)  114 ( 66)  110 ( 76)
 12   : 114 ( 75)   98 ( 71)  108 ( 61)  105 ( 91)  106 (102)  106 ( 74)   80 ( 96)  111 ( 79)
 13   : 114 ( 77)   94 ( 72)  100 ( 63)  104 ( 86)  100 (100)  107 ( 74)   71 (101)  110 ( 80)
 14   : 115 ( 77)   94 ( 75)   95 ( 66)  105 ( 79)   95 (103)  102 ( 80)   74 ( 96)  108 ( 75)
 15   : 109 ( 81)   94 ( 73)   94 ( 69)   98 ( 86)   93 (102)  102 ( 77)   77 ( 85)  112 ( 64)
 16   : 109 ( 79)   94 ( 78)   93 ( 73)   98 ( 91)   91 (102)   97 ( 78)  109 ( 64)  108 ( 74)
 17   : 111 ( 80)   96 ( 79)   93 ( 76)   99 ( 87)   93 (100)   77 ( 94)  110 ( 58)  111 ( 70)
 18   : 113 ( 79)  100 ( 73)   96 ( 71)  103 ( 81)   95 ( 96)   65 (110)  112 ( 54)  111 ( 64)
 19   : 110 ( 83)   95 ( 79)   91 ( 71)  100 ( 82)   93 ( 98)  101 ( 77)  110 ( 56)  111 ( 63)
 20   : 111 ( 84)   94 ( 79)   91 ( 71)  100 ( 84)   95 (100)  105 ( 77)  110 ( 57)  111 ( 69)
 21   : 112 ( 87)   98 ( 78)   93 ( 70)  103 ( 78)   97 ( 96)  106 ( 71)  109 ( 55)  110 ( 64)
 22   : 113 ( 89)   96 ( 79)   94 ( 67)  101 ( 79)   96 ( 98)  105 ( 69)  109 ( 58)  111 ( 70)
 23   : 117 ( 83)   98 ( 78)   92 ( 70)   97 (109)   90 (101)  101 (104)  106 ( 62)  113 ( 67)
 24   : 118 ( 73)   99 ( 70)   97 ( 65)   99 (101)   88 (111)   92 (114)  103 ( 63)  106 ( 71)
 25   : 116 ( 72)   99 ( 72)   95 ( 68)  100 (109)   85 (118)   88 (115)  100 ( 65)  104 ( 76)
 26   : 105 ( 75)   99 ( 73)   99 ( 68)  104 (101)   89 (116)   92 (118)  104 ( 63)  112 ( 67)
 27   : 110 ( 74)   97 ( 73)  101 ( 64)  107 (104)   86 (133)   92 (116)  109 ( 60)  121 ( 64)
 28   : 111 ( 77)   96 ( 78)  101 ( 63)  109 (108)   87 (131)   95 (115)  110 ( 65)  114 ( 74)
 29   : 111 ( 79)   95 ( 76)  107 ( 62)  102 (114)   85 (134)   93 (115)  106 ( 73)  108 ( 79)
 30   : 110 ( 79)   97 ( 75)  105 ( 59)  103 (116)   87 (112)   97 ( 98)  107 ( 75)  100 ( 89)
 31   : 110 ( 79)   96 ( 74)  102 ( 56)   95 (123)   86 (112)   94 (109)  105 ( 74)  108 ( 81)
 32   : 106 ( 78)   95 ( 74)   97 ( 70)   93 (124)   91 (103)   97 ( 93)  106 ( 72)  107 ( 79)
 33   : 103 ( 76)   96 ( 71)   97 ( 70)   98 (122)   91 (103)   98 (113)  107 ( 64)  111 ( 74)
 34   : 107 ( 76)   96 ( 73)   96 ( 70)   99 (114)   90 (100)   98 (116)  107 ( 66)  113 ( 63)
 35   : 106 ( 76)   98 ( 74)   97 ( 69)  102 ( 74)   93 (100)   98 (116)  116 ( 61)  115 ( 67)
 36   : 106 ( 79)   99 ( 73)  101 ( 67)  100 ( 70)   94 (103)   99 (117)  111 ( 69)  117 ( 77)
 37   : 111 ( 74)   99 ( 69)   96 ( 67)  100 (119)   91 (120)   96 (118)   77 ( 98)  114 ( 80)
 38   : 110 ( 75)   97 ( 66)   93 ( 58)   94 (123)   90 (113)   92 (118)  102 ( 76)  110 ( 86)
 39   : 112 ( 73)   92 ( 69)   86 ( 68)   92 (110)   89 (115)   89 ( 98)   93 ( 81)  109 ( 79)
 40   : 110 ( 69)   96 ( 64)   88 ( 61)   92 ( 86)   90 (104)   86 ( 91)   94 ( 82)  115 ( 77)
 41   : 115 ( 66)   96 ( 64)   90 ( 61)   94 ( 87)   89 (102)   89 (113)  101 ( 78)  115 ( 79)
 42   : 106 ( 73)   96 ( 71)   88 ( 68)   95 ( 86)   92 ( 99)   90 (114)  112 ( 69)  117 ( 78)
 43   : 103 ( 79)   96 ( 75)   92 ( 72)   94 ( 87)   90 (102)   88 (106)  109 ( 68)  114 ( 74)
 44   :  99 ( 81)   97 ( 77)   96 ( 76)   94 ( 95)   92 ( 94)   87 (117)  113 ( 66)  109 ( 79)
 45   :  98 ( 80)   99 ( 77)   99 ( 76)   95 ( 91)   90 ( 97)   89 (115)  107 ( 68)  111 ( 81)
 46   :  98 ( 82)   93 ( 81)   97 ( 78)   94 ( 93)   87 (102)   88 (125)  108 ( 64)  112 ( 73)
 47   :  92 ( 88)   94 ( 81)   96 ( 75)   91 ( 96)   87 (107)   89 (122)  111 ( 67)  116 ( 82)
 48   :  93 ( 87)   90 ( 83)   93 ( 76)   90 ( 92)   87 (109)   91 (122)  105 ( 75)  112 ( 85)
 49   :  93 ( 87)   91 ( 83)   93 ( 78)   91 ( 90)   87 (101)   92 (126)  103 ( 81)  116 ( 86)
 50   :  92 ( 87)   90 ( 84)   93 ( 77)   92 ( 88)   89 ( 97)   94 (121)  111 ( 76)  124 ( 79)
 51   :  96 ( 87)   92 ( 85)   94 ( 73)   94 ( 86)   90 ( 99)   94 (130)  101 ( 82)  116 ( 81)
 52   :  94 ( 83)   93 ( 81)   94 ( 74)   94 ( 89)   92 (102)   94 (133)   98 ( 77)  113 ( 82)
 53   :  93 ( 96)   92 ( 90)   96 ( 79)   91 ( 89)   93 (104)   95 (106)   96 ( 81)  113 ( 79)
 54   :  96 ( 84)   93 ( 77)   97 ( 75)   95 ( 83)   94 ( 99)   91 (128)   93 ( 85)  110 ( 82)
 55   :  99 ( 81)   97 ( 79)   99 ( 77)   96 ( 87)   92 (102)   87 (126)   79 ( 97)  108 ( 79)
 56   : 100 ( 82)   98 ( 78)  100 ( 72)   96 ( 75)   91 ( 96)   87 (123)   70 (101)  108 ( 73)
 57   : 103 ( 85)  103 ( 79)  100 ( 73)   96 ( 76)   89 ( 95)   84 (120)   83 (116)  109 ( 72)
 58   :  99 ( 88)  102 ( 76)  101 ( 70)   96 ( 91)   87 ( 98)   86 (121)   80 (115)  107 ( 75)
 59   : 101 ( 91)  101 ( 80)   99 ( 72)   95 ( 75)   85 ( 95)   88 (118)   94 (115)  107 ( 73)
 60   : 104 ( 98)  101 ( 83)   99 ( 75)   95 ( 78)   84 ( 99)   83 (118)   86 (101)  105 ( 67)
 61   : 106 ( 88)  104 ( 79)   99 ( 74)   89 ( 83)   83 (100)   94 (110)   92 (113)   96 ( 72)
 62   : 100 ( 88)   98 ( 79)  103 ( 71)   96 ( 77)   82 ( 97)   91 (113)   85 (115)   92 ( 75)
 63   :  93 ( 86)   96 ( 76)  100 ( 67)   97 ( 77)   84 (100)   95 (116)   79 (102)   91 ( 73)
```

```
                BRIDGE deck CONDition SUMMARY

                [ ASPhalt AVERage  THICKness ]
                          line number
                1      2      3      4      5      6      7      8
.....................................................................
         mm    107     97     97     99     92     95    103    109
         in    4.21   3.82   3.82   3.90   3.62   3.74   4.06   4.29
ave dielect    6.14   6.38   6.56   6.42   6.26   5.81   5.89   6.15

              [ C/O REINFORCEment AVErage. THICKness ]
                          line number
                1      2      3      4      5      6      7      8
.....................................................................
         mm     81     76     70     91    104    103     72     74
         in    3.19   2.99   2.76   3.58   4.09   4.06   2.83   2.91

                [ LINE DISTRIBUTion (%) ]
                          line number
                1      2      3      4      5      6      7      8
.....................................................................
       GOOD   98.76  98.14  94.39  98.13  93.46  98.44  97.50  99.07
   DEBONDed     .93    .31   1.87    .93   6.23    .62    .00    .00
     SCALEd     .00    .00    .00    .00    .00    .00    .00    .00
DELAMINATed     .31   1.55   3.74    .93    .31    .93   2.14    .31
   BAD data     .00    .00    .00    .00    .00    .00    .36    .62
      TOTAL  100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00

             metal PLATE AMPlitude is:  9.940553 v

      OVERALL AVErage  DIELECTric constant of asphalt is 6.2010

      OVERALL AVERage  ASPhalt THICKness is  99 mm, [ 3.90 in ]

      OVERALL AVErage  C/O REINFORCEment THICKness is  83 mm, [ 3.27 in ]

              [ OVERALL bridge deck CONDITION ]

                     GOOD        97.24 %
                 DEBONDED         1.36 %
                   SCALed          .00 %
              Delaminated         1.28 %
                 BAD data          .12 %
                    Total       100.00 %
```

Table 2: Survey Summary for Ninth Line Evaluated by 386-Based DART.

```
( SUMMARY )

DATE = 10 31.1990         TIME = 13:44: 4:31

NINTH LINE BRIDGE DECK site 10-02-327 north bound direction

AVERAGe  Bridge Length =   64.9 metre       NUMBER of LINES processed = 10

metre    1          2          3          4          5          6          7          8          9          10
-------------------------------------------------------------------------------------------------------------------
  1     86 ( 81)   82 ( 83)   72 ( 65)   76 ( 67)   64 ( 69)   93 ( 54)   83 ( 79)   72 ( 71)   75 ( 62)   88 ( 72)
  2    125 ( 87)   79 ( 77)  117 ( 73)  105 ( 73)  106 ( 95)  113 (127)  105 ( 79)  110 ( 70)   98 ( 71)  103 ( 79)
  3     86 ( 76)   92 ( 69)  113 ( 73)  109 ( 77)  104 (104)  111 ( 62)  108 ( 79)  114 ( 69)  100 ( 71)  103 ( 77)
  4     84 ( 77)   94 ( 68)  113 ( 75)  111 ( 75)  104 (119)  105 ( 79)  107 ( 80)  112 ( 71)  100 ( 77)  105 ( 74)
  5     83 ( 77)  121 ( 79)  103 ( 75)  107 ( 75)  106 (125)  115 ( 99)  106 ( 87)  110 ( 72)   99 ( 76)  106 ( 75)
  6    112 ( 73)  117 ( 76)  108 ( 76)  106 ( 74)  107 (110)  115 (125)  108 ( 87)  110 ( 70)   98 ( 75)  113 ( 71)
  7    113 ( 83)  112 ( 77)  106 ( 76)  101 ( 77)  103 (106)  110 ( 62)  106 ( 79)  100 ( 78)   91 ( 77)  108 ( 70)
  8    115 ( 88)  109 ( 82)  102 ( 74)  100 ( 79)   99 ( 75)  110 ( 61)  105 ( 77)   90 ( 86)   89 ( 80)  105 ( 69)
  9    104 ( 76)  107 ( 80)  103 ( 73)  100 ( 75)   98 (120)  109 ( 72)  105 ( 76)  101 ( 71)   94 ( 67)  108 ( 68)
 10    116 ( 86)  106 ( 87)  101 ( 70)   98 ( 73)   98 (111)  110 (136)  103 ( 76)  103 ( 66)   95 ( 69)  109 ( 67)
 11    114 ( 81)  107 ( 75)  103 ( 67)   98 ( 74)   95 ( 70)  111 (114)   96 ( 80)   95 ( 71)   95 ( 72)  108 ( 74)
 12    113 ( 78)  109 ( 69)  104 ( 65)  101 ( 71)   87 ( 63)  112 ( 93)  101 ( 75)   96 ( 71)   90 ( 79)  108 ( 78)
 13    114 ( 80)  108 ( 71)  102 ( 65)   99 ( 69)   84 ( 67)  110 (109)  103 ( 73)  109 ( 63)   79 ( 89)   99 ( 83)
 14    114 ( 79)  109 ( 71)   95 ( 69)   95 ( 69)   86 ( 65)  107 ( 97)  102 ( 73)  109 ( 62)   76 ( 88)  105 ( 74)
 15    110 ( 83)  110 ( 70)  101 ( 64)  100 ( 69)   91 ( 82)   93 ( 96)   97 ( 76)  104 ( 63)   88 ( 70)  109 ( 65)
 16    112 ( 83)  110 ( 74)  100 ( 69)  103 ( 68)   90 ( 85)   94 ( 96)   87 ( 83)  104 ( 63)   85 ( 78)  105 ( 75)
 17    116 ( 83)  113 ( 74)  101 ( 70)   97 ( 75)   90 ( 86)   98 ( 97)   84 ( 82)   80 ( 81)   82 ( 77)  109 ( 71)
 18    109 ( 86)  107 ( 76)   91 ( 77)   84 ( 84)   93 ( 83)   99 ( 87)   79 ( 87)   80 ( 79)   74 ( 82)   84 ( 90)
 19    107 ( 92)  111 ( 76)  102 ( 63)  100 ( 74)   91 ( 78)  101 ( 86)   86 ( 83)   74 ( 83)   95 ( 82)  114 ( 57)
 20    109 ( 95)  105 ( 82)   97 ( 68)  100 ( 68)   86 ( 87)  102 ( 86)   97 ( 72)  104 ( 60)   77 ( 81)   94 ( 83)
 21    110 ( 93)  108 ( 83)  101 ( 63)  103 ( 77)   90 ( 82)  107 ( 85)   91 ( 76)  103 ( 57)   89 ( 67)  102 ( 73)
 22    116 ( 89)  113 ( 78)  100 ( 64)   95 ( 71)   85 ( 85)   85 ( 92)   86 ( 78)   89 ( 71)   82 ( 74)  114 ( 68)
 23    112 ( 90)   98 ( 91)   90 ( 75)   99 ( 83)   89 ( 75)   87 ( 66)   84 ( 77)   83 ( 79)   94 ( 64)  115 ( 66)
 24     83 ( 71)  110 ( 70)   90 ( 72)  101 ( 88)   93 ( 90)   88 ( 64)   89 ( 76)   98 ( 65)   90 ( 64)  109 ( 71)
 25    114 ( 75)   99 ( 80)   94 ( 80)  100 ( 92)   82 ( 81)   85 ( 69)   84 ( 76)   97 ( 62)   85 ( 71)  108 ( 73)
 26    107 ( 76)   93 ( 80)   85 ( 81)   95 (104)   79 ( 76)   87 ( 64)   86 ( 66)   98 ( 73)   88 ( 64)  116 ( 64)
 27    102 ( 86)   95 ( 82)   86 ( 80)   90 (113)   85 ( 71)   80 ( 70)   87 ( 68)  101 ( 64)   81 ( 74)  123 ( 60)
 28    106 ( 87)  108 ( 75)   89 ( 77)   90 (113)   84 ( 64)   84 ( 69)   90 ( 66)  105 ( 63)   72 ( 90)  114 ( 73)
 29    108 ( 84)  108 ( 75)   94 ( 76)   86 (109)   81 ( 72)   90 ( 64)   85 ( 69)  100 ( 74)   93 ( 77)  105 ( 80)
 30    109 ( 84)  111 ( 73)   93 ( 74)   96 ( 99)   83 ( 70)   91 ( 63)   90 ( 72)  103 ( 75)   93 ( 77)   94 ( 89)
 31    108 ( 84)  110 ( 73)   97 ( 65)  108 (101)   89 ( 69)   93 ( 64)   90 ( 63)  104 ( 74)   91 ( 79)  103 ( 82)
 32    108 ( 79)  108 ( 71)   98 ( 72)  105 (109)   81 ( 70)  100 ( 61)   95 ( 65)  101 ( 73)   94 ( 75)  104 ( 82)
 33    107 ( 79)  107 ( 71)   99 ( 71)  105 (110)   89 ( 70)   96 ( 61)   94 ( 66)  104 ( 62)   96 ( 68)  105 ( 78)
 34    107 ( 81)  112 ( 70)  101 ( 70)  110 (113)   89 ( 68)   95 ( 62)   95 ( 62)  109 ( 69)  100 ( 92)  112 ( 68)
 35    108 ( 82)  109 ( 72)  103 ( 70)  110 (121)   92 ( 61)   95 ( 60)   95 ( 63)  111 ( 60)  104 ( 65)  117 ( 70)
 36    111 ( 83)  108 ( 71)  104 ( 70)  107 (123)   92 ( 65)   96 ( 60)   93 ( 66)  105 ( 68)  105 ( 72)  112 ( 85)
 37    110 ( 81)  108 ( 69)   98 ( 73)   98 (120)   90 ( 74)   94 ( 61)   93 ( 82)  105 ( 67)  103 ( 72)  105 ( 90)
 38    105 ( 80)  104 ( 71)   97 ( 62)  103 (102)   84 ( 92)   93 ( 64)   87 ( 95)  103 ( 64)  101 ( 73)  100 ( 97)
 39    106 ( 84)  103 ( 69)   88 ( 68)   99 ( 93)   82 ( 99)   92 ( 63)   85 ( 90)   99 ( 70)   96 ( 71)  102 ( 86)
 40    107 ( 80)  106 ( 64)   93 ( 94)   97 ( 68)   82 ( 89)   91 ( 97)   82 ( 86)   93 ( 73)   96 ( 72)  109 ( 80)
 41    110 ( 77)  110 ( 62)   94 ( 97)   97 ( 69)   83 ( 87)   92 ( 96)   87 ( 78)   93 ( 75) [illegible] ( 71)  111 ( 85)
 42    108 ( 81)  107 ( 71)   88 ( 66)   98 ( 73)   85 ( 87)   91 ( 97)   88 ( 77)   96 ( 71)   99 ( 74)  114 ( 80)
 43    105 ( 82)  104 ( 73)   89 ( 70)  100 ( 74)   83 ( 87)   92 ( 94)   86 ( 75)   95 ( 70)   99 ( 70)  111 ( 76)
 44     79 ( 90)  103 ( 75)   96 ( 74)   96 ( 77)   83 ( 86)   94 ( 95)   87 ( 67)   92 ( 70)   99 ( 71)  109 ( 82)
 45     79 (103)  105 ( 75)   98 ( 74)  100 ( 77)   86 ( 85)   94 ( 88)   82 ( 69)   95 ( 62)   98 ( 69)  111 ( 79)
 46     80 (101)  106 ( 75)  102 ( 74)  102 ( 78)   84 ( 89)   91 ( 94)   81 ( 77)   93 ( 65)   98 ( 66)  111 ( 77)
 47    111 ( 77)  105 ( 78)   96 ( 77)  101 ( 81)   82 ( 94)   90 ( 98)   82 ( 77)   91 ( 69)  101 ( 66)  112 ( 72)
 48    108 ( 81)  103 ( 79)   93 ( 75)   96 ( 80)   83 ( 93)   91 ( 84)   84 ( 78)   97 ( 63)  102 ( 70)  113 ( 85)
 49    108 ( 82)   99 ( 80)   90 ( 75)   96 ( 77)   82 ( 92)   90 ( 89)   94 ( 83)   98 ( 68)  102 ( 72)   76 ( 79)
 50    108 ( 79)   99 ( 78)   92 ( 75)   95 ( 76)   82 ( 90)   92 ( 95)   92 ( 75)   97 ( 76)  102 ( 77)   80 ( 77)
 51    110 ( 77)  100 ( 79)   90 ( 76)   97 ( 74)   84 ( 87)   93 ( 93)   95 ( 74)  103 ( 70)  105 ( 74)   81 ( 76)
 52    109 ( 77)   98 ( 79)   94 ( 72)   98 ( 71)   83 ( 84)   92 ( 96)   91 ( 79)  102 ( 72)  101 ( 75)  118 ( 81)
 53    110 ( 82)   98 ( 81)   95 ( 71)   98 ( 76)   83 ( 90)   94 ( 91)   89 ( 83)   97 ( 75)   99 ( 75)  112 ( 85)
 54    111 ( 80)   97 ( 80)   94 ( 78)  101 ( 78)   78 ( 98)   98 (103)   88 ( 80)   94 ( 74)   99 ( 74)  111 ( 82)
 55    112 ( 76)  100 ( 76)  100 ( 73)  106 ( 74)   79 ( 95)   95 ( 96)   85 ( 80)   90 ( 77)   91 ( 84)  111 ( 83)
 56    107 ( 81)   98 ( 78)   96 ( 75)  105 ( 73)   83 ( 91)   94 ( 93)   85 ( 77)   89 ( 76)   93 ( 78)  112 ( 75)
 57    112 ( 80)  106 ( 78)   99 ( 70)  103 ( 67)   89 ( 79)   91 ( 90)   85 ( 74)   90 ( 73)   88 ( 77)  113 ( 69)
 58    108 ( 83)  109 ( 78)   95 ( 72)  100 ( 65)   87 ( 79)   89 ( 91)   86 ( 73)   91 ( 69)   82 ( 83)  116 ( 66)
 59    109 ( 92)  100 ( 85)   99 ( 70)  104 ( 65)   95 ( 77)   91 ( 90)   83 ( 78)   86 ( 80)   80 ( 82)  115 ( 70)
 60    113 ( 95)  106 ( 85)   96 ( 74)  102 ( 67)   96 ( 76)   90 ( 92)   83 ( 79)   77 ( 84)   79 ( 80)  113 ( 71)
 61    113 ( 91)  108 ( 81)   94 ( 78)   79 ( 90)   84 ( 87)   61 ( 85)   81 ( 83)   81 ( 81)   74 ( 84)  105 ( 72)
 62    107 ( 87)  106 ( 81)   97 ( 74)   74 ( 93)   94 ( 77)   74 ( 85)   84 ( 82)   77 ( 83)   74 ( 84)   94 ( 75)
 63    108 ( 87)   99 ( 78)   98 ( 74)  100 ( 79)   91 ( 82)   68 ( 85)   85 ( 84)   80 ( 84)   82 ( 77)   90 ( 76)
 64     99 ( 81)  102 ( 73)   93 ( 73)   98 ( 80)   85 ( 89)   86 ( 80)   81 ( 88)   71 ( 89)   76 ( 74)   89 ( 71)
```

```
( SUMMARY )

NINTH LINE BRIDGE DECK site 10-02-327 north bound direction

DATE = 10 31.1990         TIME = 13:44:10:53

( FLAT PLATE AMPLITUDE )  =    7.926542 v
( FIFTH WHEEL )           =   10.000000 cm
( ASPHALT  core sample )  =   100.0 mm
( CONCRETE core sample )  =    85.0 mm

( AVERAGe  DIELECTRIC CONSTANT OF ASPHALT )

                 1      2      3      4      5      6      7      8      9     10
-----------------------------------------------------------------------------------
               6.65   6.55   6.95   7.01   7.73   6.34   6.61   6.39   7.56   6.88

( AVERAGe  ASPHALT THICKNESS )

                 1      2      3      4      5      6      7      8      9     10
-----------------------------------------------------------------------------------
millimetres    106    104     97     99     88     95     91     96     91    106

( AVERAGe  CONCRETE COVER )

                 1      2      3      4      5      6      7      8      9     10
-----------------------------------------------------------------------------------
millimetres     83     76     73     83     85     84     77     72     75     76

( PERCENTAGE OF DAMAGES )

                 1      2      3      4      5      6      7      8      9     10

  SCALed        .00    .62   9.05    .00   5.97   9.12   2.63    .00    .00    .00
DELAMINATed     .15    .77   1.99    .31   7.20   6.96   7.26   7.06   3.24   3.09
 DEBONDed       .00   1.39   2.61    .15  20.98   9.43    .00    .00    .15    .62
-----------------------------------------------------------------------------------
  DAMAGED       .15   2.77  13.65    .46  34.15  25.50   9.89   7.06   3.39   3.70

( CONDITIONS )

                 1      2      3      4      5      6      7      8      9     10
-----------------------------------------------------------------------------------
  DAMAGED       .15   2.77  13.65    .46  34.15  25.50   9.89   7.06   3.39   3.70
   GOOD       99.85  97.23  86.35  99.54  65.85  74.50  90.11  92.94  96.61  96.30
 CORRUPled      .00    .00    .00    .00    .00    .00    .00    .00    .00    .00
-----------------------------------------------------------------------------------
  TOTAL      100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00

( OVERALL AVERAGES )

( dielectric constant of aspahlt ) =  6.866960

( asphalt thickness (mm) )          :  97

( concrete cover (mm) )             :  78

( PERCENTAGE of DAMAGES )

SCALED        2.74 %
DELAMINATED   3.80 %
DEBONDED      3.53 %
---------------------
DAMAGED      10.07 %

( CONDITIONS )

DAMAGED      10.07 %
GOOD         89.93 %
CORRUPTED      .00 %
---------------------
TOTAL       100.00 %
```

Table 3: Survey for Ninth Line from Penetradar's Radar Inspection.

9th LINE BRIDGE
ASPHALT AND CONCRETE COVER THICKNESS SUMMARY

	LINE		1		2		3		4		5		6		7		8		9		10			
	PASS - 1		PASS - 2		PASS - 3		PASS - 4		PASS - 5		PASS - 6		PASS - 7		PASS - 8		PASS - 9		PASS - 10		PASS - 11		PASS - 12	
METER	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE
1	90	101	105	87	105	73	95	83	86	83	90	92	76	110	86	92	95	69	76	69	90	73	81	73
2	105	87	109	83	109	73	105	64	95	73	95	92	86	110	95	87	95	69	86	69	95	73	81	73
3	109	83	114	73	109	78	105	73	105	73	105	92	90	110	105	83	95	69	95	69	100	73	81	73
4	114	78	109	87	109	73	105	64	105	73	105	87	95	96	100	78	95	69	95	73	100	78	86	69
5	114	73	105	87	100	78	95	78	105	69	105	83	105	87	95	87	95	69	90	78	100	78	86	64
6	114	73	100	78	100	73	95	73	95	69	100	87	100	87	90	83	95	73	86	78	100	73	81	78
7	109	83	100	87	100	78	95	73	105	64	95	96	100	87	90	83	81	78	76	83	100	64	76	92
8	109	87	105	87	95	78	105	64	95	73	90	92	100	101	95	78	81	73	81	78	100	64	90	87
9	109	87	105	87	95	78	95	64	90	73	90	87	100	110	95	78	76	73	86	73	100	64	105	83
10	100	83	105	83	90	73	90	69	105	64	86	96	105	101	100	78	71	69	90	69	100	69	105	83
11	105	83	100	78	95	64	90	69	90	78	86	101	105	92	105	92	81	69	95	64	100	73	105	87
12	109	69	105	78	90	73	90	69	90	78	90	92	100	92	76	92	86	64	86	73	95	73	105	87
13	100	92	100	78	100	69	71	78	90	60	100	83	95	87	95	83	95	64	81	83	95	78	105	87
14	105	78	105	73	100	64	90	64	105	64	90	87	95	83	90	78	95	60	86	78	95	78	109	83
15	109	83	100	78	105	69	95	60	109	83	86	92	90	83	86	69	95	55	90	69	100	73	114	78
16	114	73	100	87	109	64	95	69	95	69	86	92	86	87	86	83	81	78	76	73	100	69	114	83
17	105	78	109	69	100	69	100	64	105	60	86	92	81	92	86	96	67	83	71	78	105	64	109	92
18	114	83	105	73	105	69	86	69	109	50	86	92	86	92	86	83	76	78	86	73	100	64	109	83
19	95	96	100	83	105	64	86	73	114	73	90	92	86	92	86	73	90	64	95	64	100	69	114	73
20	100	96	100	87	109	64	90	64	114	78	90	92	90	87	86	69	90	64	95	64	100	69	109	73
21	105	101	105	83	100	69	81	78	100	69	90	73	90	92	90	64	86	69	95	64	100	69	109	78
22	114	78	105	83	109	69	71	87	90	64	90	78	86	92	86	83	81	64	90	64	95	69	100	83
23	105	87	109	73	109	55	90	69	109	78	90	87	81	92	86	92	76	64	90	64	95	64	95	87
24	100	83	109	78	100	64	76	78	90	87	86	96	76	92	86	96	81	60	90	64	100	64	100	83
25	105	83	105	64	105	46	76	83	81	110	76	101	71	96	81	101	86	60	90	64	100	69	105	78
26	90	60	105	69	114	55	76	83	81	115	76	96	71	96	81	96	90	64	81	73	95	73	109	78
27	90	60	109	69	100	69	81	73	81	115	76	92	76	96	81	106	95	64	71	83	95	78	114	73
28	109	78	100	78	90	83	81	78	76	110	76	92	81	92	81	110	86	69	81	83	90	78	109	78
29	100	87	105	78	100	73	81	73	76	110	76	92	90	92	86	110	81	69	86	78	86	83	105	83
30	100	87	100	83	95	73	81	69	71	110	81	96	90	92	81	110	86	69	90	78	90	87	109	87
31	100	83	100	83	95	69	81	78	71	110	86	101	90	92	76	110	90	69	90	83	95	87	109	87
32	105	83	100	73	95	69	81	83	71	106	90	96	90	92	81	110	90	69	95	87	100	78	109	87
33	109	78	100	78	95	69	90	73	71	96	90	92	90	92	86	110	95	64	95	96	100	73	105	87
34	109	83	105	78	95	69	86	73	86	83	86	101	90	92	86	110	100	64	100	96	105	69	105	92
35	105	87	109	69	95	69	90	73	100	64	81	110	90	92	81	110	86	64	100	101	109	64	105	92
36	100	87	105	73	86	78	81	78	90	83	81	101	86	96	81	92	86	64	100	83	100	87	100	92
37	105	92	100	78	95	69	86	69	86	96	86	92	86	96	81	96	90	69	100	64	95	92	95	92
38	109	92	100	78	86	78	71	83	90	83	91	87	81	96	71	87	90	64	95	69	100	83	100	92
39	109	92	100	73	95	64	86	64	90	64	81	87	81	101	76	83	90	64	90	73	105	73	109	87
40	109	87	105	73	100	64	86	60	95	64	86	92	81	101	76	78	86	64	95	69	105	78	109	87
41	95	92	100	69	95	69	81	69	95	69	90	92	81	106	76	73	81	69	95	64	105	83	114	92
42	109	78	105	78	86	73	86	69	95	73	86	92	86	96	81	73	76	73	95	69	95	78	109	92
43	114	73	100	73	86	78	90	73	90	73	86	92	86	87	81	69	81	69	90	69	100	73	105	92
44	105	73	100	69	95	69	90	78	90	69	95	92	81	92	76	78	81	64	90	64	95	78	105	92
45	109	69	105	73	90	73	95	73	90	69	86	87	76	92	71	87	76	73	95	60	95	78	[illegible]	92
46	109	78	95	78	90	78	90	78	86	69	86	87	86	92	76	78	76	69	100	60	105	73	114	96
47	105	78	90	83	86	83	90	73	86	64	86	92	90	96	81	69	81	64	100	64	109	73	114	96
48	100	87	90	87	90	73	86	73	90	73	90	92	90	96	86	92	81	69	100	69	114	73	[illegible]	96
49	109	78	90	83	90	73	86	78	90	78	[illegible]	92	86	92	90	101	81	73	95	78	114	73	[illegible]	92
50	100	78	90	87	90	73	86	78	95	87	86	92	81	92	86	101	95	73	95	73	109	78	114	[illegible]
51	100	78	95	83	81	83	90	73	95	96	81	92	81	96	81	106	90	73	95	73	105	83	116	96
52	100	78	95	83	86	83	90	73	95	92	81	96	86	96	81	101	86	78	95	73	100	83	109	[illegible]
53	105	83	95	87	81	69	90	78	95	87	81	92	90	92	81	92	81	78	95	73	95	87	109	[illegible]
54	105	83	100	78	90	78	95	73	100	87	81	92	86	92	81	106	76	69	95	73	95	83	114	92
55	100	87	100	83	90	78	90	73	100	83	81	87	86	92	81	110	76	69	90	73	95	83	119	87
56	100	87	105	78	95	73	95	69	95	83	81	83	81	92	81	110	81	69	90	73	95	78	119	83
57	100	87	100	83	100	73	90	69	95	83	81	73	81	92	81	110	81	69	86	78	95	73	114	73
58	105	87	95	87	95	73	95	69	95	78	86	83	86	92	81	110	76	69	86	73	95	69	114	73
59	105	92	100	83	100	78	90	69	95	69	86	87	86	87	81	106	71	87	86	69	105	64	109	78
60	100	92	100	92	100	73	86	83	81	78	76	101	100	83	71	110	67	87	71	73	90	69	100	87
61	109	78	95	87	105	69	90	73	76	83	67	110	109	73	67	110	67	87	62	78	81	73	95	96
62	95	87	86	87	90	73	90	73	71	78	71	101	100	87	71	110	62	83	67	73	81	69	95	87
63	90	83	90	78	86	78	86	69	67	73	76	92	81	96	81	64	57	83	76	73	81	64	90	78
64	86	73	86	83	71	83	76	73	67	69	81	96	81	92	86	46	57	92	81	78	86	69	86	73

9th LINE BRIDGE
THICKNESS SUMMARY

AVERAGE ASPHALT THICKNESS

PASS	1	2	3	4	5	6	7	8	9	10	11	12	OVERALL
MM	104	101	96	88	91	86	88	84	83	89	98	104	93
INCHES	4.09	3.97	3.78	3.47	3.59	3.37	3.45	3.30	3.26	3.49	3.85	4.11	3.65
Line		1	2	3	4	5	6	7	8	9	10		

AVERAGE CONCRETE REBAR COVER

PASS	1	2	3	4	5	6	7	8	9	10	11	12	OVERALL
MM	83	80	72	73	80	92	93	91	70	73	74	85	80
INCHES	3.25	3.13	2.82	2.86	3.13	3.61	3.67	3.57	2.76	2.89	2.92	3.34	3.16

Note:
Penetradar's
Pass 2 - 11
correspond to
DART's Line 1 - 10

Table 3: (continue) Survey for Ninth Line from Penetradar's Radar Inspection.

```
NINTH LINE BRIDGE OVER HIGHWAY 403

RADAR INSPECTION DATE: JUNE 14, 1990

DECK CONSTRUCTION: Reinforced concrete with asphalt overlay.  Two
traffic lanes and shoulders.

RADAR RESULTS: OVERALL PERCENT OF BRIDGE DECK DETERIORATED

          3.2 % ------- DELAMINATION
          9.4 % ------- DEBONDING
          0.0 % ------- SCALING

*************************************************************
                       D E L A M I N A T I O N

PASS NUMBER  1  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  2  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  3  HAS   5  DETECTIONS OUT OF   130  FOR   3.8  %
PASS NUMBER  4  HAS   5  DETECTIONS OUT OF   130  FOR   3.8  %
PASS NUMBER  5  HAS   8  DETECTIONS OUT OF   130  FOR   6.2  %
PASS NUMBER  6  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  7  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  8  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  9  HAS  10  DETECTIONS OUT OF   130  FOR   7.7  %
PASS NUMBER 10  HAS  14  DETECTIONS OUT OF   130  FOR  10.8  %
PASS NUMBER 11  HAS   2  DETECTIONS OUT OF   130  FOR   1.5  %
PASS NUMBER 12  HAS   6  DETECTIONS OUT OF   130  FOR   4.6  %

9th Line Bridge is  3.20512821 % Delaminated by Area

*************************************************************

                        D E B O N D I N G

PASS NUMBER  1  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  2  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  3  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  4  HAS  15  DETECTIONS OUT OF   130  FOR  11.5  %
PASS NUMBER  5  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  6  HAS  42  DETECTIONS OUT OF   130  FOR  32.3  %
PASS NUMBER  7  HAS  49  DETECTIONS OUT OF   130  FOR  37.7  %
PASS NUMBER  8  HAS  30  DETECTIONS OUT OF   130  FOR  23.1  %
PASS NUMBER  9  HAS   2  DETECTIONS OUT OF   130  FOR   1.5  %
PASS NUMBER 10  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER 11  HAS   6  DETECTIONS OUT OF   130  FOR   4.6  %
PASS NUMBER 12  HAS   2  DETECTIONS OUT OF   130  FOR   1.5  %

9th Line Bridge is  9.35897436 % Debonded by Area

*************************************************************

                          S C A L I N G

PASS NUMBER  1  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  2  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  3  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  4  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  5  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  6  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  7  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  8  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER  9  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER 10  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER 11  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %
PASS NUMBER 12  HAS   0  DETECTIONS OUT OF   130  FOR   0.0  %

9th Line Bridge is  0.0 % Scaled by Area

*************************************************************

VISUAL OBSERVATIONS:  Bridge deck top and bottom surfaces in very
good condition.  No water stains or cracks observed on bottom
surface of deck.  Asphalt in very good condition, no signs of
asphalt deterioration.

COMMENTS:  Radar inspection detected a very slight amount of
delamination, which was more prevalent in south lane but not con-
fined to any particular region of the bridge deck.  Debonded
asphalt detected primarily about the center line of the bridge
deck between 34 and 57 meters in passes F,G and H.  No scaling of
concrete at asphalt/concrete interface was found.

Overall bridge deck condition is very good.
```

Note:
Penetradar's
Pass 2 - 11
correspond to
DART's
Line 1 - 10.

Table 4: Survey Summary for Little Bob South Bridge.

```
[ SUMMARY ]

DATE = 10 31.1990          TIME = 14:18:59:72

LITTLE BOB: SOUTH BRIDGE site 32-079, North bound direction

AVERAGe  Bridge Length =   24.2 metre      NUMBER of LINES processed = 10

metre    1           2           3           4           5           6           7           8           9           10
   ----------------------------------------------------------------------------------------------------------------------
  1    96 (  71)  101 (  75)   95 (  85)  102 ( 102)   94 (  96)   91 (  81)   94 ( 114)  141 (  61)  154 (  47)  136 (  41)
  2   122 (  75)  126 (  88)  113 ( 106)  113 ( 107)  108 (  94)  101 (  92)  105 ( 131)   92 ( 131)   94 (  99)  143 (  52)
  3   135 (  73)  149 (  83)  124 ( 108)  123 ( 111)  119 (  89)  108 (  87)  107 ( 139)   98 ( 125)   97 (  94)  142 (  59)
  4   137 (  71)  154 (  80)  128 ( 105)  132 ( 101)  124 (  86)  111 (  79)  114 ( 108)  102 ( 106)  104 (  87)  104 (  93)
  5   131 (  64)  149 (  71)  126 (  83)  129 (  79)  125 (  71)  111 (  62)  117 (  77)   98 (  76)  102 (  65)  102 (  64)
  6   136 (  45)  142 (  49)  126 (  65)  126 (  52)  126 (  47)  110 (  49)  117 (  53)  101 (  47)  107 (  45)  107 (  46)
  7   131 (  88)  141 (  70)  133 (  56)  136 (  44)  138 (  60)  130 (  49)  128 (  66)  110 (  77)  114 (  65)  120 (  85)
  8   129 (  76)  137 (  76)  134 (  53)  133 (  45)  145 (  79)  146 (  62)  143 (  50)  121 (  49)  125 (  91)  129 (  85)
  9   131 (  65)  142 (  69)  143 (  46)  141 (  48)  156 (  78)  100 (  42)  148 (  62)  125 (  55)  140 (  87)  133 (  79)
 10   136 (  66)  142 (  62)  141 (  41)  136 (  47)  153 (  75)  151 (  66)  142 (  64)  120 (  57)  136 (  92)  138 (  75)
 11   126 (  64)  138 (  51)  133 (  41)  135 (  46)  149 (  76)  136 (  63)  134 (  72)  111 (  89)  122 (  77)  126 (  77)
 12   117 (  73)  137 (  50)  121 (  77)  126 (  47)  138 (  77)  119 (  65)  118 (  43)   97 (  51)  105 (  42)  101 (  74)
 13   113 (  63)  122 (  48)  113 (  47)  116 (  49)  128 (  75)  116 (  63)  114 (  56)   95 (  50)   98 (  49)   95 (  79)
 14   114 (  79)  120 (  51)  113 (  51)  120 (  45)  126 (  82)  128 (  56)  124 (  51)  107 (  41)  105 (  85)   99 (  87)
 15   119 (  76)  131 (  56)  121 (  65)  126 (  45)  128 (  71)  134 (  55)  129 (  55)  110 (  65)  114 (  88)  112 (  93)
 16   116 (  83)  136 (  54)  121 (  63)  123 (  45)  127 (  64)  134 (  48)  132 (  81)  115 (  49)  120 (  99)  120 (  89)
 17   117 (  95)  129 (  70)  119 (  65)  124 (  45)  127 (  72)  135 (  67)  134 (  47)  118 (  43)  115 (  76)  121 (  86)
 18   114 (  92)  123 (  75)  113 (  61)  119 (  56)  124 (  73)  121 (  76)  124 (  64)  111 (  58)  112 (  67)  117 (  83)
 19   112 (  48)  118 (  51)  110 (  52)  113 (  44)  117 (  49)  107 (  73)  121 (  71)  108 (  55)  108 (  52)  115 (  79)
 20   113 (  56)  123 (  65)  112 (  59)  114 (  56)  109 (  59)  114 (  78)  121 (  91)  107 (  77)  104 (  45)  101 (  42)
 21   116 (  64)  131 (  73)  119 (  77)  124 (  69)  124 (  55)  120 (  80)  124 ( 111)  107 (  93)  108 (  65)  106 (  69)
 22   116 (  64)  133 (  76)  118 (  80)  122 (  83)  122 (  57)  127 (  80)  129 ( 115)  113 (  96)  101 (  77)  106 (  78)
 23   112 (  63)  122 (  78)  111 (  88)  117 (  93)  111 (  68)  120 (  92)  125 ( 115)  107 ( 105)   96 (  80)   98 (  85)
 24   101 (  99)  112 (  75)  100 (  89)  103 ( 103)  101 (  87)  108 ( 108)  116 ( 113)  999 ( 999)   94 (  75)   98 (  64)
```

```
[ SUMMARY ]

LITTLE BOB: SOUTH BRIDGE site 32-079, North bound direction

DATE = 10 31.1990          TIME = 14:19: 1:14

[ FLAT PLATE AMPLITUDE ] =    9.381371 v
[ FIFTH WHEEL ]          :   10.000000 cm
[ ASPHALT  core sample ] =  120.0 mm
[ CONCRETE core sample ] =   60.0 mm

[ AVERAGe  DIELECTRIC CONSTANT OF ASPHALT ]

               1      2      3      4      5      6      7      8      9     10
-------------------------------------------------------------------------------
             5.24   4.98   5.57   5.61   5.40   5.43   5.33   5.84   5.45   5.37

[ AVERAGe  ASPHALT THICKNESS ]

               1      2      3      4      5      6      7      8      9     10
-------------------------------------------------------------------------------
millimetres  120    131    120    123    125    120    123    109    112    115

[ AVERAGe  CONCRETE COVER ]

               1      2      3      4      5      6      7      8      9     10
-------------------------------------------------------------------------------
millimetres   70     67     69     66     73     70     81     73     73     73

[ PERCENTAGE OF DAMAGES ]

               1      2      3      4      5      6      7      8      9     10
-------------------------------------------------------------------------------
   SCALed    19.34  22.22   8.64  18.52  13.17  10.37  28.51  60.43  19.92  10.37
DELAMINATed  21.81  19.34  32.92  52.26  34.57  32.78  21.07  17.02  29.83  27.39
 DEBONDed      .82   4.53    .41   9.05    .41    .00   2.48   2.55   7.47    .00
-------------------------------------------------------------------------------
 DAMAGED     41.98  46.09  41.98  79.84  48.15  43.15  52.07  80.00  57.26  37.76

[ CONDITIONS ]

               1      2      3      4      5      6      7      8      9     10
-------------------------------------------------------------------------------
 DAMAGED     41.98  46.09  41.98  79.84  48.15  43.15  52.07  80.00  57.26  37.76
 GOOD        58.02  53.91  58.02  20.16  51.85  56.85  47.93  20.00  42.74  62.24
CORRUPTed      .00    .00    .00    .00    .00    .00    .00    .00    .00    .00
-------------------------------------------------------------------------------
 TOTAL      100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00

[ OVERALL AVERAGES ]

[ dielectric constant of asphalt ] :  5.422378

[ asphalt thickness (mm) ]         :  119

[ concrete cover (mm) ]            :  71

[ PERCENTAGE of DAMAGES ]

SCALED        21.15 %
DELAMINATED   28.90 %
DEBONDED       2.77 %
----------------------
DAMAGED       52.83 %

[ CONDITIONS ]

DAMAGED       52.83 %
GOOD          47.17 %
CORRUPTED       .00 %
----------------------
TOTAL        100.00 %
```

Table 5: Survey for Little Bob South Bridge from Penetradar's Radar Inspection.

Line	5		4		3		2		1	
	PASS - E		PASS - D		PASS - C		PASS - B		PASS - A	
	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE
	2.7	[illegible]	3.1	4.60	[illegible]	3.76	[illegible]	[illegible]	4.6	2.72
	2.6	3.97	3.2	4.60	3.2	4.39	3.2	3.55	4.6	2.93
	3.1	3.55	3.6	4.81	3.5	4.60	3.7	3.55	4.9	3.14
	3.6	3.55	3.0	4.60	3.7	4.60	3.8	3.55	5.2	3.14
	3.6	3.34	3.7	3.97	3.9	4.39	3.8	3.14	4.7	3.14
	3.7	2.51	3.5	2.51	3.5	3.14	3.7	2.09	4.6	2.51
	4.0	2.09	3.8	2.30	3.9	2.72	3.5	2.72	4.7	2.51
	4.1	2.30	3.8	2.09	4.0	2.72	3.5	2.93	4.6	3.14
	4.5	2.09	4.0	2.09	4.0	2.09	3.7	2.93	4.5	3.14
	4.8	2.09	4.4	2.09	4.2	2.30	3.6	3.97	4.8	3.55
	4.7	2.30	4.0	2.72	4.0	1.88	3.8	3.76	4.0	2.51
	4.6	2.93	3.9	1.67	3.7	1.67	4.0	3.14	4.5	2.93
	4.0	2.09	3.5	2.30	3.7	1.88	3.6	3.55	2.9	3.76
	4.0	2.30	3.5	2.30	3.5	2.09	3.3	4.18	3.8	3.34
	4.5	2.72	3.7	2.30	3.7	1.25	3.7	3.55	3.9	3.34
	4.1	3.34	3.8	1.88	3.8	1.05	3.8	3.97	3.9	3.55
	4.0	2.51	3.9	1.88	3.8	1.46	3.6	4.18	3.9	3.14
	4.8	3.34	3.6	2.30	3.6	2.30	3.5	4.18	4.0	2.30
	3.6	2.30	3.4	1.46	3.3	3.34	3.2	4.60	4.0	2.09
	3.4	2.51	3.2	2.30	3.3	3.55	3.3	2.93	3.8	3.14
	3.5	2.72	3.4	2.93	3.7	3.14	3.6	3.14	3.9	3.14
	3.8	2.30	3.6	3.55	3.6	3.55	3.6	3.34	3.8	3.34
	3.5	2.72	3.4	3.97	3.7	3.76	3.2	3.55	3.7	3.14
	3.5	3.14	3.2	4.39	3.3	3.97	3.2	3.55	3.5	2.93
	3.87	2.77	3.63	2.90	3.64	2.90	3.50	3.45	3.90	3.02

[illegible]
ASPHALT AND CONCRETE COVER THICKNESS SUMMARY

Line	10		9		8		7		6	
	PASS - J		PASS - I		PASS - H		PASS - G		PASS - F	
METERS	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE
1	[illegible]	[illegible]	2.6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2.4	[illegible]
2	3.0	1.9	2.2	[illegible]	2.4	5.23	[illegible]	5.43	3.1	4.18
3	3.2	1.7	3.3	4.39	3.0	5.23	3.1	5.64	[illegible]	4.18
4	2.6	1.9	3.5	4.39	3.0	4.18	3.4	5.64	3.2	4.18
5	2.9	1.7	3.4	3.76	3.2	4.39	3.3	3.97	2.5	3.55
6	2.9	1.7	3.4	2.72	2.8	2.93	3.5	2.51	2.7	2.30
7	3.3	2.3	3.5	3.34	3.2	2.09	3.7	1.88	3.2	2.09
8	3.4	2.1	4.3	2.72	3.5	2.09	4.2	2.09	3.0	2.09
9	2.7	2.5	4.7	2.51	3.9	2.30	4.4	2.51	3.3	2.51
10	2.7	2.5	3.4	2.72	4.0	2.72	4.4	2.93	3.5	2.51
11	2.9	2.1	4.4	2.30	3.7	2.09	4.2	1.88	3.2	2.30
12	2.7	1.9	4.0	1.67	3.2	1.46	3.7	1.88	3.8	3.55
13	2.7	1.7	3.3	1.88	2.8	2.09	3.3	2.51	3.9	4.18
14	2.4	1.9	3.3	2.09	2.9	2.09	3.5	2.51	4.1	3.76
15	3.0	1.9	4.0	2.51	4.3	1.88	4.5	1.46	3.7	3.76
16	2.8	2.1	3.5	2.51	3.5	1.46	4.0	1.46	4.0	3.76
17	2.7	2.5	3.7	2.93	3.5	1.88	4.1	1.46	4.0	3.55
18	2.4	2.5	4.3	2.72	5.0	2.30	4.5	2.30	3.8	3.97
19	2.6	2.5	3.5	2.72	3.3	2.09	3.5	1.46	2.8	3.76
20	2.5	2.1	3.1	3.97	3.3	2.72	3.5	3.76	3.2	3.76
21	2.7	2.3	3.2	2.72	3.3	3.97	3.0	3.55	3.2	3.76
22	2.8	2.1	3.4	3.34	3.4	4.18	3.7	5.02	3.2	3.97
23	2.8	1.7	3.3	3.97	3.3	4.60	4.0	5.02	2.7	3.97
24	2.6	1.9	3.2	3.76	3.2	4.60	3.7	5.02	2.5	4.60
AVG	2.78	2.03	3.56	3.12	3.35	3.04	3.71	3.30	3.26	3.47

AVERAGE ASPHALT THICKNESS

PASS	J	I	H	G	F	E	D	C	B	A	OVERALL
MM	71	90	85	94	83	98	92	92	89	[illegible]	88.4
INCHES	2.78	3.56	3.35	3.71	3.26	3.87	3.63	3.64	3.50	[illegible]	3.48

AVERAGE CONCRETE REBAR COVER

LINE	10	9	8	7	6	5	4	3	2	1	
PASS	J	I	H	G	F	E	D	C	B	A	OVERALL
MM	52	79	77	84	88	70	74	74	88	77	75.9
INCHES	2.03	3.12	3.04	3.30	3.47	2.77	2.90	2.90	3.45	3.02	2.99

Table 5: (continue) Survey for Little Bob South Bridge from Penetradar's Radar Inspection.

```
*********************************************
M.T.O. BRIDGE 32-79 - INCREASE IN A/C MAP #2

PASS NUMBER  1  HAS  19  DETECTIONS OUT OF  48
PASS NUMBER  2  HAS  4  DETECTIONS OUT OF  48
PASS NUMBER  3  HAS  10  DETECTIONS OUT OF  48
PASS NUMBER  4  HAS  1  DETECTIONS OUT OF  48
PASS NUMBER  5  HAS  5  DETECTIONS OUT OF  48
PASS NUMBER  6  HAS  2  DETECTIONS OUT OF  48
PASS NUMBER  7  HAS  10  DETECTIONS OUT OF  48
PASS NUMBER  8  HAS  11  DETECTIONS OUT OF  48
PASS NUMBER  9  HAS  26  DETECTIONS OUT OF  48
PASS NUMBER  10  HAS  14  DETECTIONS OUT OF  48

THANK YOU FOR WAITING

M.T.O. BRIDGE 32-79 - INCREASE IN A/C MAP #2 IS  21.4583333  % DETERIORATED BY A
REA

*********************************************

*********************************************
M.T.O. - BRIDGE 32-79 - W DETECTION

PASS NUMBER  1  HAS  8  DETECTIONS OUT OF  48
PASS NUMBER  2  HAS  7  DETECTIONS OUT OF  48
PASS NUMBER  3  HAS  6  DETECTIONS OUT OF  48
PASS NUMBER  4  HAS  4  DETECTIONS OUT OF  48
PASS NUMBER  5  HAS  14  DETECTIONS OUT OF  48
PASS NUMBER  6  HAS  4  DETECTIONS OUT OF  48
PASS NUMBER  7  HAS  9  DETECTIONS OUT OF  48
PASS NUMBER  8  HAS  9  DETECTIONS OUT OF  48
PASS NUMBER  9  HAS  8  DETECTIONS OUT OF  48
PASS NUMBER  10  HAS  21  DETECTIONS OUT OF  48

THANK YOU FOR WAITING

M.T.O. - BRIDGE 32-79 - W DETECTION IS  18.9583333  % DETERIORATED BY AREA

*********************************************
```

Table 6: Survey Summary for Little Bob North Bridge.

```
[ SUMMARY ]

DATE = 10 31.1990          TIME = 14:54:31:10

LITTLE BOB NORTH BRIDGE site 32-080, north bound direction

AVERAGe Bridge Length =   23.8 metre      NUMBER of LINES processed = 10

metre   1        2        3        4        5        6        7        8        9        10
     ---------------------------------------------------------------------------------------
  1  [illegible]
  2  [illegible]
  3  [illegible]
  4  [illegible]
  5  [illegible]
  6  [illegible]
  7  [illegible]
  8  [illegible]
  9  [illegible]
 10  [illegible]
 11  [illegible]
 12  [illegible]
 13  [illegible]
 14  [illegible]
 15  [illegible]
 16  [illegible]
 17  [illegible]
 18  [illegible]
 19  [illegible]
 20  [illegible]
 21  [illegible]
 22  [illegible]
 23  [illegible]
```

```
[ SUMMARY ]

LITTLE BOB NORTH BRIDGE site 32-080, north bound direction

DATE = 10 31.1990          TIME = 14:54:32:47

[ FLAT PLATE AMPLITUDE ]  =    9.880410 v
[ FIFTH WHEEL ]           =   10.000000 cm
[ ASPHALT  core sample ]  =  115.0 mm
[ CONCRETE core sample ]  =   60.0 mm

[ AVERAGe DIELECTRIC CONSTANT OF ASPHALT ]

               1      2      3      4      5      6      7      8      9      10
--------------------------------------------------------------------------------
             5.16   4.80   5.63   5.32   5.38   5.19   5.08   5.41   5.30   5.10

[ AVERAG. ASPHALT THICKNESS ]

               1      2      3      4      5      6      7      8      9      10
--------------------------------------------------------------------------------
millimetres   123    126    109    116    104    110    125    114    112    108

[ AVERAGe CONCRETE COVER ]

               1      2      3      4      5      6      7      8      9      10
--------------------------------------------------------------------------------
millimetres    77     85     89     90     96     71     85     89     74     74

[ PERCENTAGE OF DAMAGES ]

               1      2      3      4      5      6      7      8      9      10
--------------------------------------------------------------------------------
  SCALed     37.18  28.45  22.46  25.63  27.50  17.57  33.61  18.91  19.58  24.17
DELAMINATed  14.53  12.13  17.80  18.91  14.58  19.25   9.24   3.78  22.08  11.67
 DEBONDed     8.55    .42    .00  34.45   2.50    .00    .00    .00    .00  10.83
--------------------------------------------------------------------------------
 DAMAGED     60.26  41.00  40.25  78.99  44.58  36.82  42.86  22.69  41.67  46.67

[ CONDITIONS ]

               1      2      3      4      5      6      7      8      9      10
--------------------------------------------------------------------------------
 DAMAGED     60.26  41.00  40.25  78.99  44.58  36.82  42.86  22.69  41.67  46.67
   GOOD      39.74  59.00  59.75  21.01  55.42  63.18  57.14  77.31  58.33  53.33
 CORRUPTed     .00    .00    .00    .00    .00    .00    .00    .00    .00    .00
--------------------------------------------------------------------------------
  TOTAL     100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00

[ OVERALL AVERAGES ]

[ dielectric constant of asphalt ] =  5.23944

[ asphalt thickness (mm) ]         = 115

[ concrete cover (mm) ]            = 82

[ PERCENTAGE of DAMAGES ]

SCALED       25.51 %
DELAMINATED  14.40 %
DEBONDED      5.68 %
-------------------
DAMAGED      45.58 %

[ CONDITIONS ]

DAMAGED      45.58 %
GOOD         54.42 %
CORRUPTED      .00 %
-------------------
TOTAL       100.00 %
```

Table 7: Survey for Little Bob North Bridge from Penetradar's Radar Inspection.

Line	5		4		3		2		1	
	PASS - E		PASS - D		PASS - C		PASS - B		PASS - A	
	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE
	2.3	4.39	2.5	4.39	2.7	2.93	2.7	3.76	3.6	2.51
	2.4	4.39	3.0	5.23	2.9	5.23	2.8	4.18	4.3	2.93
	2.6	4.39	2.8	5.85	2.8	4.18	3.0	4.81	5.6	2.93
	2.6	4.18	2.8	5.85	3.0	4.18	3.5	4.39	5.2	2.72
	2.6	4.39	2.7	5.23	3.0	3.97	3.7	4.18	5.2	2.51
	2.4	2.30	3.1	4.18	3.3	3.97	3.5	3.34	5.0	2.51
	2.6	2.93	3.4	3.97	3.7	3.14	3.8	3.76	4.7	3.14
	3.3	3.14	3.4	3.55	3.7	2.93	3.9	3.55	5.3	3.14
	3.5	3.76	3.6	2.51	4.0	2.51	4.2	3.55	5.4	2.72
	3.5	2.30	3.5	2.93	3.8	2.72	4.0	3.97	5.1	2.93
	2.4	3.34	3.8	3.34	3.7	2.51	4.0	3.97	5.2	2.93
	2.8	2.51	3.5	3.34	3.7	2.93	4.0	3.97	5.4	3.14
	3.6	2.09	3.7	2.93	3.7	3.14	3.9	3.55	5.5	2.93
	3.6	2.51	3.7	2.51	3.5	2.51	4.0	3.55	5.4	3.34
	3.7	1.88	3.5	2.09	3.5	2.30	4.0	3.97	5.4	3.34
	3.9	2.30	3.4	1.88	3.4	2.30	3.8	4.18	4.7	3.55
	3.4	2.30	3.8	2.09	3.3	2.09	3.6	4.18	5.0	3.55
	3.3	1.88	3.5	2.30	3.3	2.30	3.6	3.76	5.1	2.09
	3.2	3.55	3.6	3.14	3.3	3.34	3.7	2.09	4.2	2.30
	3.1	3.97	3.5	4.18	3.5	3.34	3.5	3.97	4.9	3.76
	5.0	1.88	3.3	5.02	3.3	3.55	3.5	4.81	4.6	4.18
	3.0	4.60	3.0	5.43	3.0	3.76	3.1	5.23	5.2	3.97
	2.6	4.18	2.7	5.23	2.8	4.18	3.0	4.81	3.7	4.18
	2.5	3.55	2.5	4.81	2.6	4.18	2.8	3.97	4.7	2.51
	3.08	3.20	3.25	3.83	3.31	3.26	3.57	3.98	4.93	3.07

MTO BRIDGE 33-80 BOBCAYGEON
ASPHALT AND CONCRETE COVER THICKNESS SUMMARY

Line	10		9		8		7		6	
	PASS - J		PASS - I		PASS - H		PASS - G		PASS - F	
METER	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE	ASPHALT	CONCRETE
1 :	3.7	2.5	2.3	3.34	2.5	2.93	3.0	3.14	2.6	2.72
2 :	4.2	3.6	2.9	4.18	3.2	4.39	3.6	4.39	3.1	3.55
3 :	3.4	4.2	3.5	4.39	3.2	5.43	3.8	3.14	3.0	3.97
4 :	4.2	4.2	3.3	4.39	3.4	5.23	3.8	3.34	3.2	4.18
5 :	4.6	4.2	3.1	4.39	3.6	4.60	4.0	5.02	2.5	3.76
6 :	4.1	3.8	3.4	3.34	3.6	4.39	3.6	3.55	2.7	2.72
7 :	5.0	4.4	3.4	4.18	3.6	3.55	4.0	2.93	3.2	2.72
8 :	5.3	4.2	4.0	2.30	3.8	2.72	4.0	2.51	3.0	3.34
9 :	4.8	4.6	3.8	2.51	3.5	2.72	5.0	3.14	3.3	2.51
10 :	5.0	4.2	3.6	2.51	3.4	3.14	4.5	3.14	3.5	2.51
11 :	6.1	4.6	3.6	2.72	3.6	2.72	4.3	2.93	3.2	2.09
12 :	4.2	4.2	3.8	2.51	5.0	2.93	4.8	3.14	3.8	2.51
13 :	4.8	4.2	3.5	2.30	3.8	3.34	4.8	2.72	3.9	2.51
14 :	6.4	4.2	3.5	3.97	3.5	3.34	4.7	2.51	4.1	2.51
15 :	6.3	4.4	3.5	3.14	3.5	3.97	4.7	2.51	3.7	2.51
16 :	3.8	2.5	3.5	3.14	3.8	3.76	5.0	2.72	4.0	3.14
17 :	4.0	3.8	3.6	3.14	3.7	3.55	4.5	2.51	4.0	2.51
18 :	4.0	4.4	3.5	2.93	3.7	3.76	4.5	3.97	3.8	4.18
19 :	4.6	3.8	3.4	2.30	3.3	4.18	3.9	3.55	2.8	2.51
20 :	4.5	3.8	3.4	3.76	3.3	4.60	3.8	3.14	3.2	2.93
21 :	4.7	3.8	3.4	4.39	3.3	5.85	3.7	3.34	3.2	3.76
22 :	3.6	3.8	2.9	4.18	3.2	5.02	3.9	3.34	3.2	3.76
23 :	2.4	3.8	2.9	4.18	3.0	5.02	3.4	5.02	2.7	4.18
24 :	3.0	3.6	2.9	3.55	2.8	4.18	3.0	4.18	2.5	3.97
AVG	4.40	3.93	3.36	3.40	3.47	3.97	4.10	3.33	3.26	3.13

MTO BRIDGE 33-80 BOBCAYGEON
THICKNESS SUMMARY

AVERAGE ASPHALT THICKNESS

PASS	1	2	3	4	5	6	7	8	9	10	11	12	OVERALL
MM	112	85	88	104	83	78	83	84	91	125			93.3
INCHES	4.40	3.36	3.47	4.10	3.26	3.08	3.25	3.31	3.57	4.93			3.67

AVERAGE CONCRETE REBAR COVER

PASS	1	2	3	4	5	6	7	8	9	10	11	12	OVERALL
MM	100	86	101	84	79	81	97	83	101	78			89.1
INCHES	3.93	3.40	3.97	3.33	3.13	3.20	3.83	3.26	3.98	3.07			3.51

Table 7: (continue) Survey for Little Bob North Bridge from Penetradar's Radar Inspection.

```
********************************************
M.T.O. BRIDGE 32-80 - INCREASE IN A/C MAP #2

PASS NUMBER  1  HAS  14  DETECTIONS OUT OF  48
PASS NUMBER  2  HAS  16  DETECTIONS OUT OF  48
PASS NUMBER  3  HAS  6  DETECTIONS OUT OF  48
PASS NUMBER  4  HAS  22  DETECTIONS OUT OF  48
PASS NUMBER  5  HAS  15  DETECTIONS OUT OF  48
PASS NUMBER  6  HAS  9  DETECTIONS OUT OF  48
PASS NUMBER  7  HAS  1  DETECTIONS OUT OF  48
PASS NUMBER  8  HAS  4  DETECTIONS OUT OF  48
PASS NUMBER  9  HAS  13  DETECTIONS OUT OF  48
PASS NUMBER  10  HAS  24  DETECTIONS OUT OF  48

THANK YOU FOR WAITING

M.T.O. BRIDGE 32-80 - INCREASE IN A/C MAP #2 IS  26.0416667  % DETERIORATED BY A
REA

********************************************
```

```
********************************************
M.T.O. - BRIDGE 32-80 - W DETECTION

PASS NUMBER  1  HAS  15  DETECTIONS OUT OF  48
PASS NUMBER  2  HAS  12  DETECTIONS OUT OF  48
PASS NUMBER  3  HAS  18  DETECTIONS OUT OF  48
PASS NUMBER  4  HAS  4  DETECTIONS OUT OF  48
PASS NUMBER  5  HAS  5  DETECTIONS OUT OF  48
PASS NUMBER  6  HAS  7  DETECTIONS OUT OF  48
PASS NUMBER  7  HAS  1  DETECTIONS OUT OF  48
PASS NUMBER  8  HAS  3  DETECTIONS OUT OF  48
PASS NUMBER  9  HAS  12  DETECTIONS OUT OF  48
PASS NUMBER  10  HAS  19  DETECTIONS OUT OF  48

THANK YOU FOR WAITING

M.T.O. - BRIDGE 32-80 - W DETECTION IS  20  % DETERIORATED BY AREA

********************************************
```

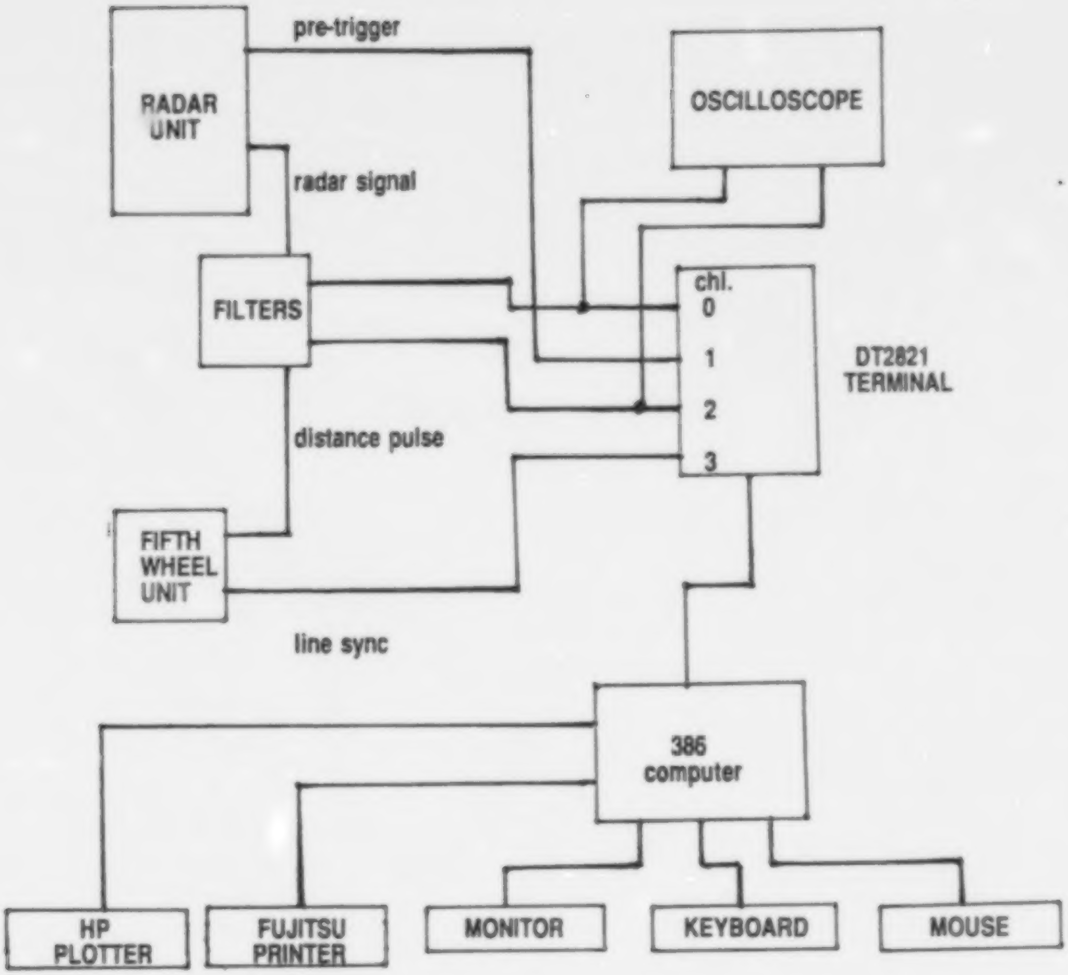

RADAR UNIT
pre-trigger
radar signal
OSCILLOSCOPE
FILTERS
chl.
0
1
2
3
DT2821 TERMINAL
distance pulse
FIFTH WHEEL UNIT
line sync
386 computer
HP PLOTTER
FUJITSU PRINTER
MONITOR
KEYBOARD
MOUSE

Figure 1: System Configuration Design for 386-Based DART.

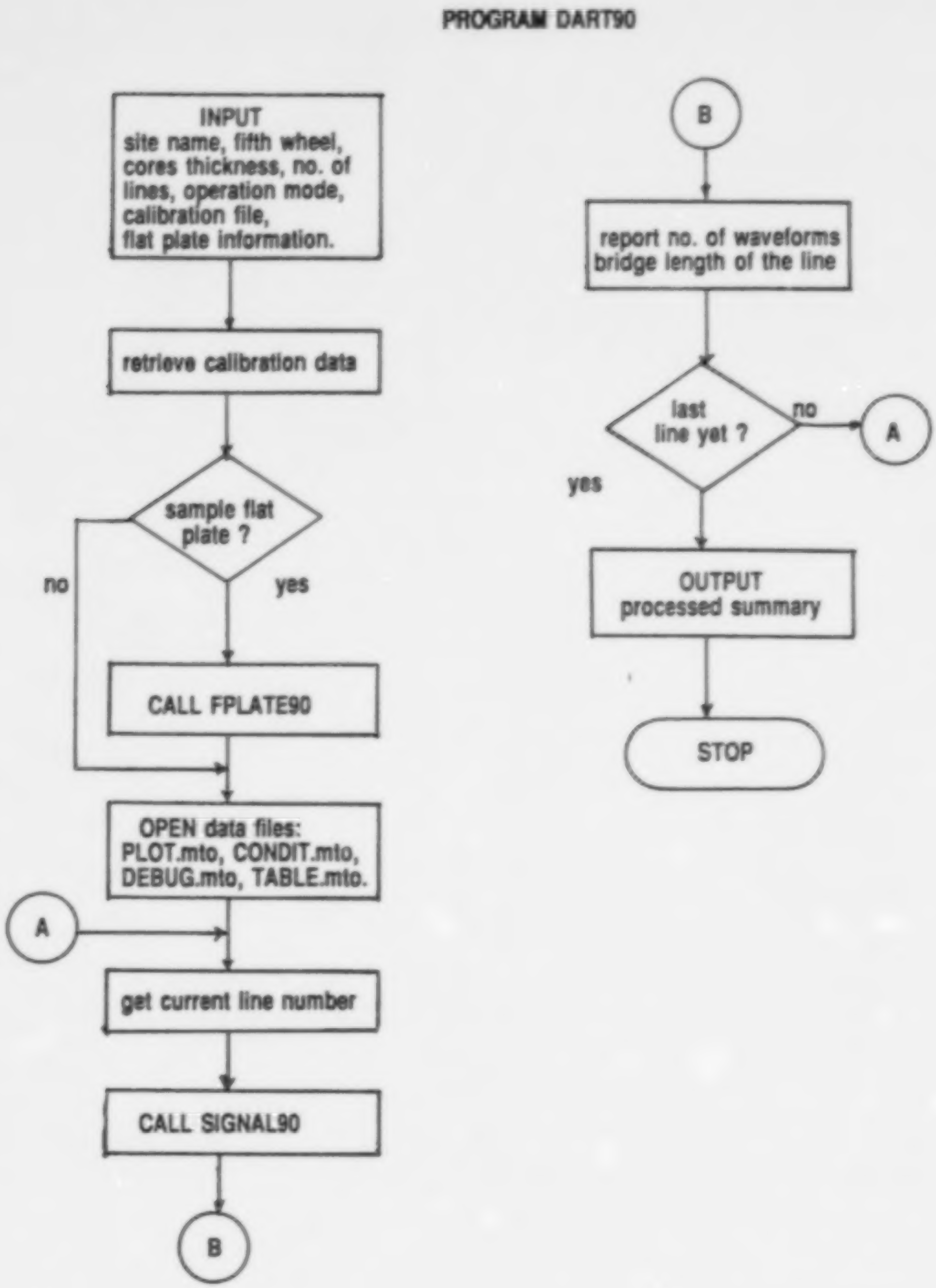

PROGRAM DART90
INPUT
site name, fifth wheel, cores thickness, no. of lines, operation mode, calibration file, flat plate information.
retrieve calibration data
sample flat plate ?
no
yes
CALL FPLATE90
OPEN data files: PLOT.mto, CONDIT.mto, DEBUG.mto, TABLE.mto.
A
get current line number
CALL SIGNAL90
B
B
report no. of waveforms bridge length of the line
last line yet ?
no
A
yes
OUTPUT
processed summary
STOP

Figure 2: Flow Diagram for Main Program DART90

```
DATE = 10 4.1990      TIME = 15:51:33:56

[ DART SOFTWARE PACKAGE ]

- type BRIDGE/SITE name -
example: bridge name or site number                    (a)
```

```
DATE = 10 4.1990      TIME = 15:51:33:56

[ DART SOFTWARE PACKAGE ]

example: bridge name or site number

-fifth wheel information -
type (0)-10 cm, (1)-1 m, (2)-10 m : 0                  (b)
```

```
DATE = 10 4.1990      TIME = 15:52: 9:76

[ DART SOFTWARE PACKAGE ]

example: bridge name or site number

- ASPHALT core information -
give thickness of asphalt core (mm) : 100

- CONCRETE cover information -
input(0)-automatic, (1)-operator input : 1

give thickness of concrete cover (mm) : 90             (c)
```

```
DATE = 10 4.1990      TIME = 15:52:21:84

[ DART SOFTWARE PACKAGE ]

example: bridge name or site number

number of lines (max 20) to be collected : 2

[ mode ] 0 - ANALOG, 1 - data files : 0

[ DEBUG ] 0-no, 1-yes : 0

[ CALIBRATION ] data file name (*.MTO) : cal90.mto     (d)
```

```
DATE = 10 4.1990      TIME = 15:52:36:84

[ DART SOFTWARE PACKAGE ]

example: bridge name or site number

[ FLAT PLATE ] (0)-analog, (1)-file, (2)-input : 0     (e)
```

Figure 3: Sample Inputs for DART90 Software.

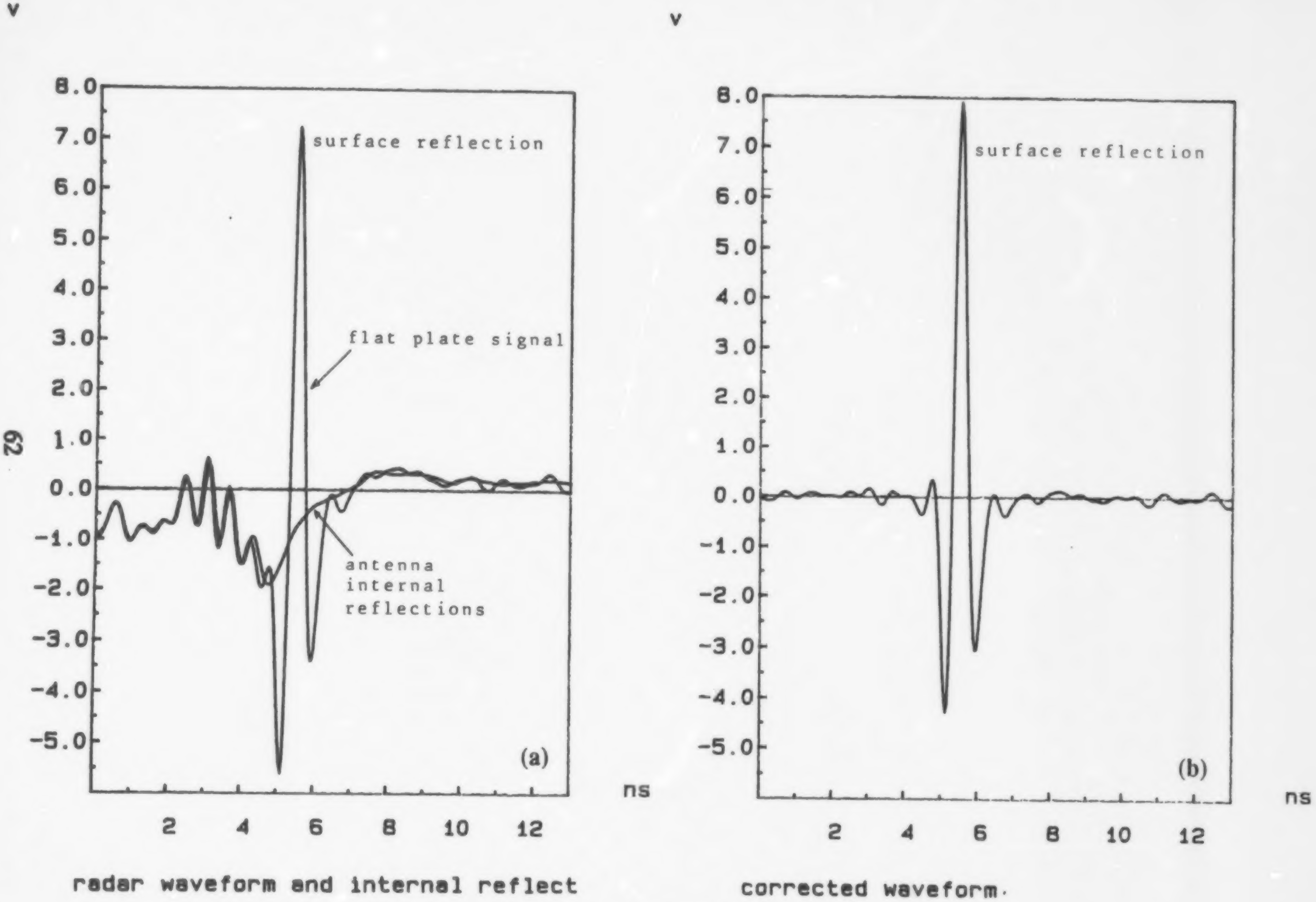


Figure 4: (a) Superimposed Waveforms from Flat Plate Signal and Internal Antenna Reflections. (b) Corrected Flat Plate Signal.

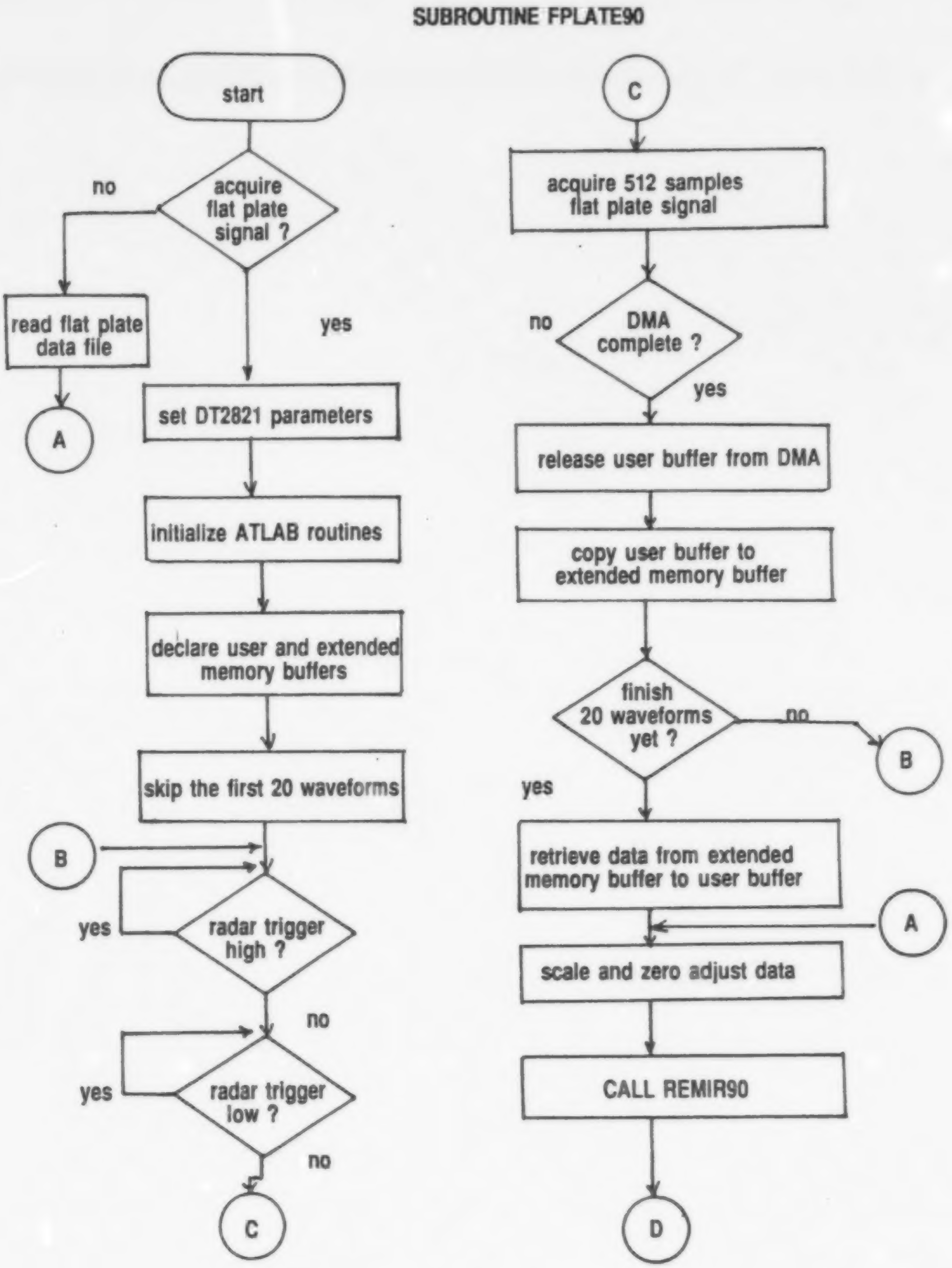

SUBROUTINE FPLATE90
start
acquire flat plate signal ?
no
yes
read flat plate data file
A
set DT2821 parameters
initialize ATLAB routines
declare user and extended memory buffers
skip the first 20 waveforms
B
yes
radar trigger high ?
no
yes
radar trigger low ?
no
C
C
acquire 512 samples flat plate signal
no
DMA complete ?
yes
release user buffer from DMA
copy user buffer to extended memory buffer
finish 20 waveforms yet ?
no
B
yes
retrieve data from extended memory buffer to user buffer
A
scale and zero adjust data
CALL REMIR90
D

Figure 5: Flow Diagram for Subroutine FPLATE90

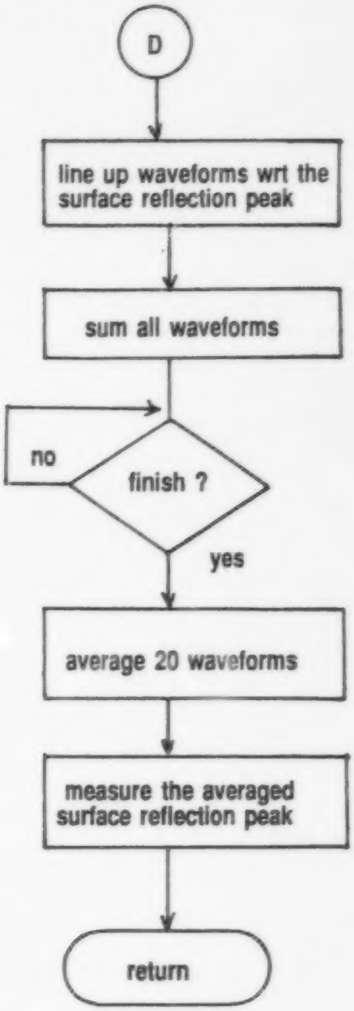

D
line up waveforms wrt the surface reflection peak
sum all waveforms
no
finish ?
yes
average 20 waveforms
measure the averaged surface reflection peak
return

Figure 5: (continue) Flow Diagram for Subroutine FPLATE90

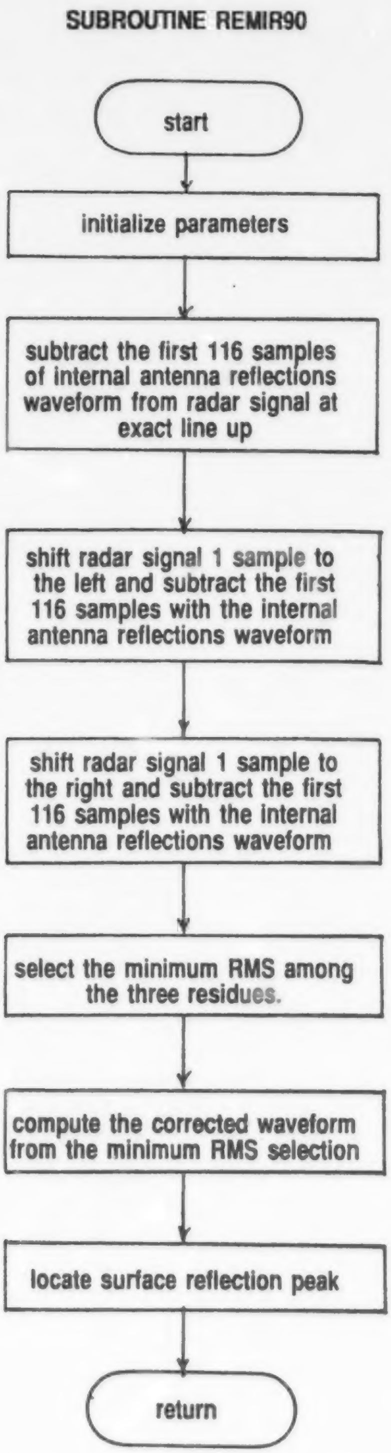

SUBROUTINE REMIR90
start
initialize parameters
subtract the first 116 samples of internal antenna reflections waveform from radar signal at exact line up
shift radar signal 1 sample to the left and subtract the first 116 samples with the internal antenna reflections waveform
shift radar signal 1 sample to the right and subtract the first 116 samples with the internal antenna reflections waveform
select the minimum RMS among the three residues.
compute the corrected waveform from the minimum RMS selection
locate surface reflection peak
return

Figure 6: Flow Diagram for Subroutine REMIR90

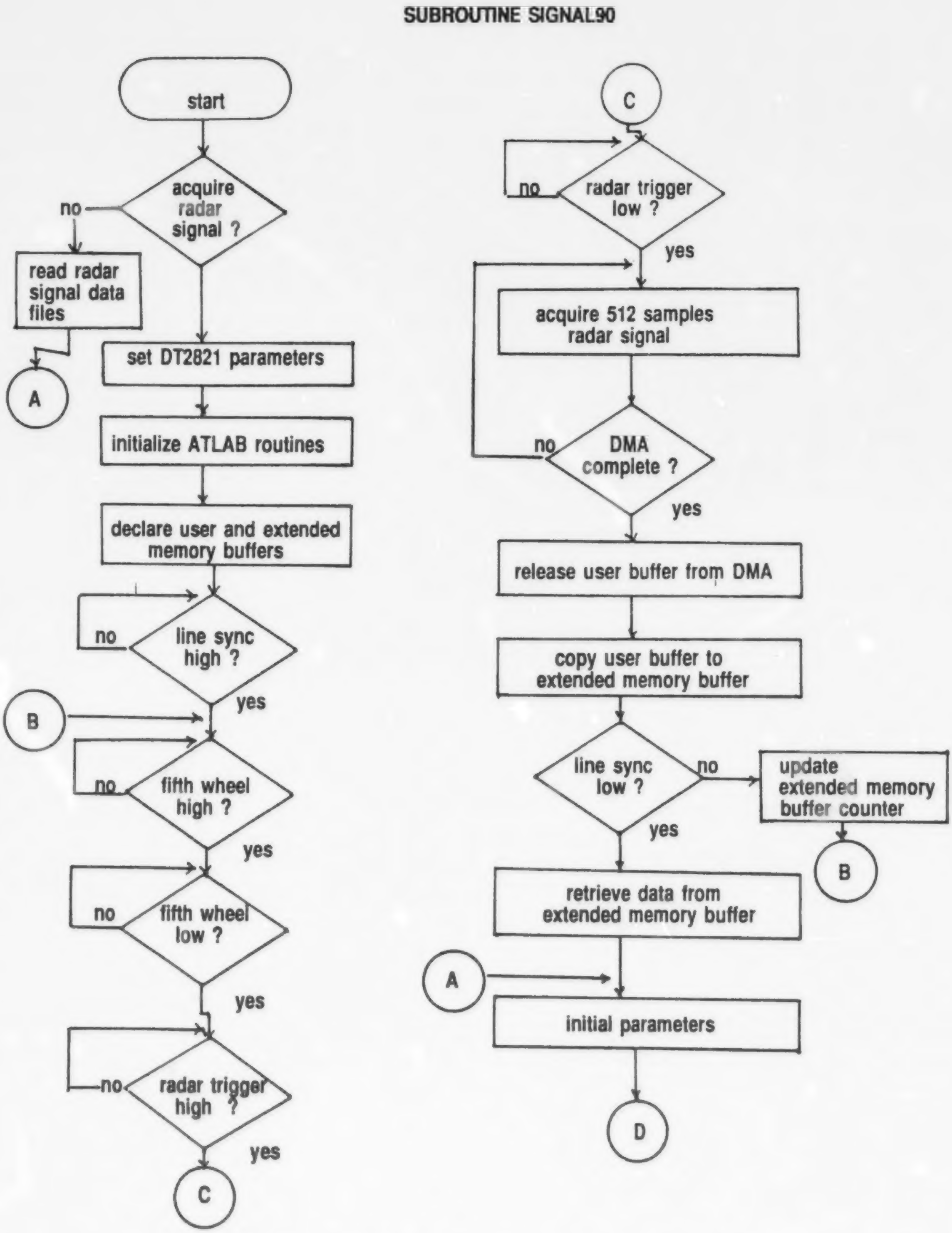

SUBROUTINE SIGNAL90
start
acquire radar signal ?
no
read radar signal data files
A
set DT2821 parameters
initialize ATLAB routines
declare user and extended memory buffers
no
line sync high ?
yes
B
no
fifth wheel high ?
yes
no
fifth wheel low ?
yes
no
radar trigger high ?
yes
C
C
no
radar trigger low ?
yes
acquire 512 samples radar signal
no
DMA complete ?
yes
release user buffer from DMA
copy user buffer to extended memory buffer
line sync low ?
no
update extended memory buffer counter
B
yes
retrieve data from extended memory buffer
A
initial parameters
D

Figure 7: Flow Diagram for Subroutine SIGNAL90

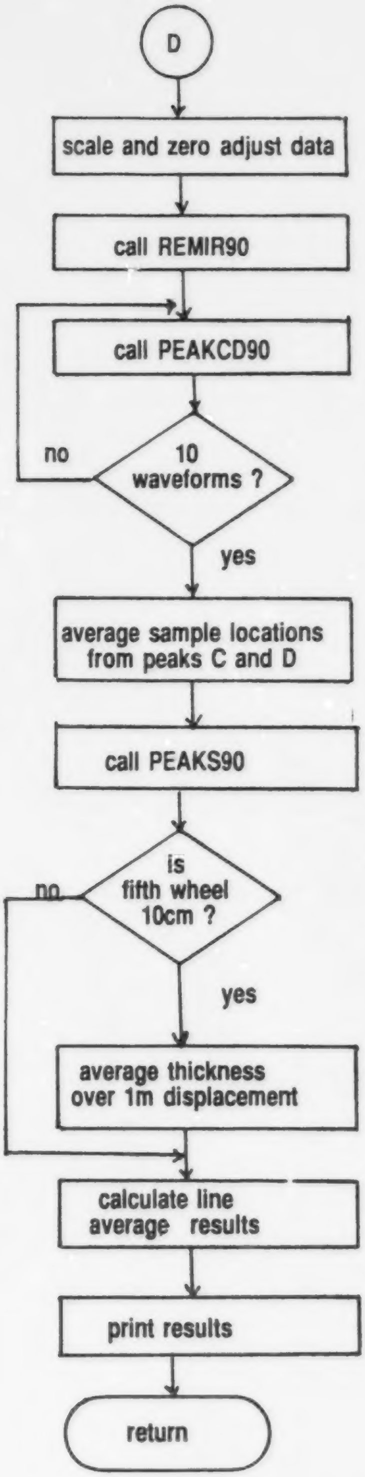

D
scale and zero adjust data
call REMIR90
call PEAKCD90
no
10 waveforms ?
yes
average sample locations from peaks C and D
call PEAKS90
no
is fifth wheel 10cm ?
yes
average thickness over 1m displacement
calculate line average results
print results
return

Figure 7: (continued) Flow Diagram for Subroutine SIGNAL90

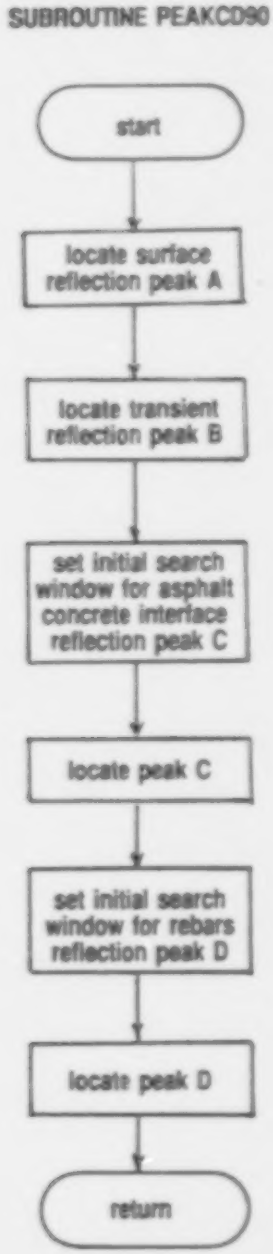

SUBROUTINE PEAKCD90
start
locate surface reflection peak A
locate transient reflection peak B
set initial search window for asphalt concrete interface reflection peak C
locate peak C
set initial search window for rebars reflection peak D
locate peak D
return

Figure 8: Flow Diagram for Subroutine PEAKCD90

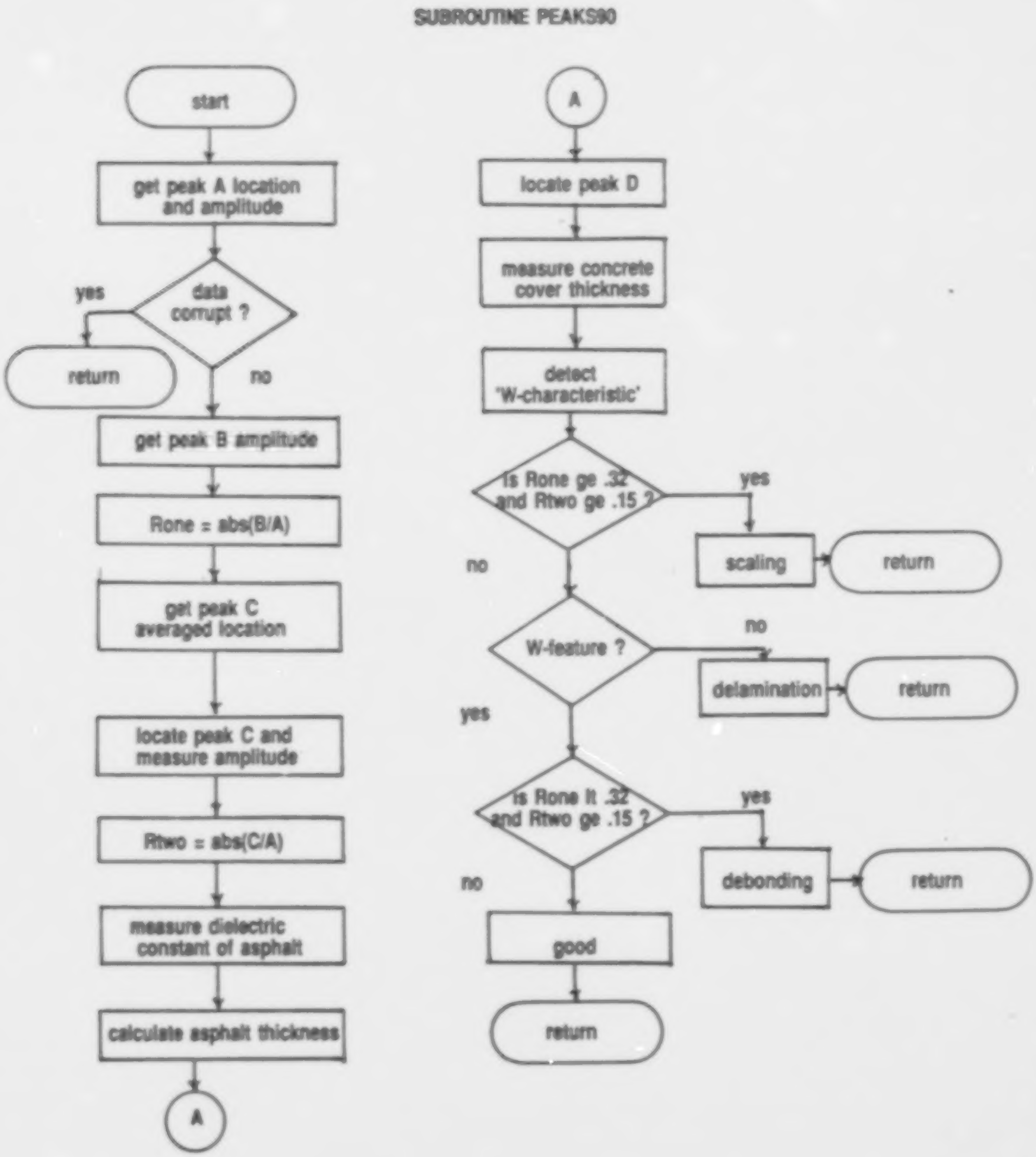

SUBROUTINE PEAKS90
start
get peak A location and amplitude
data corrupt ?
yes
return
no
get peak B amplitude
Rone = abs(B/A)
get peak C averaged location
locate peak C and measure amplitude
Rtwo = abs(C/A)
measure dielectric constant of asphalt
calculate asphalt thickness
A
A
locate peak D
measure concrete cover thickness
detect 'W-characteristic'
is Rone ge .32 and Rtwo ge .15 ?
yes
scaling
return
no
W-feature ?
no
delamination
return
yes
is Rone lt .32 and Rtwo ge .15 ?
yes
debonding
return
no
good
return

Figure 9: Flow Diagram for Subroutine PEAKS90

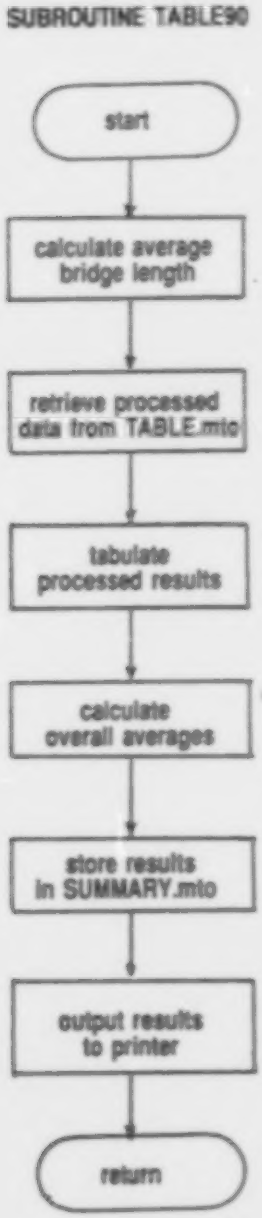

SUBROUTINE TABLE90
start
calculate average bridge length
retrieve processed data from TABLE.mto
tabulate processed results
calculate overall averages
store results in SUMMARY.mto
output results to printer
return

Figure 10: Flow Diagram for Subroutine TABLE90

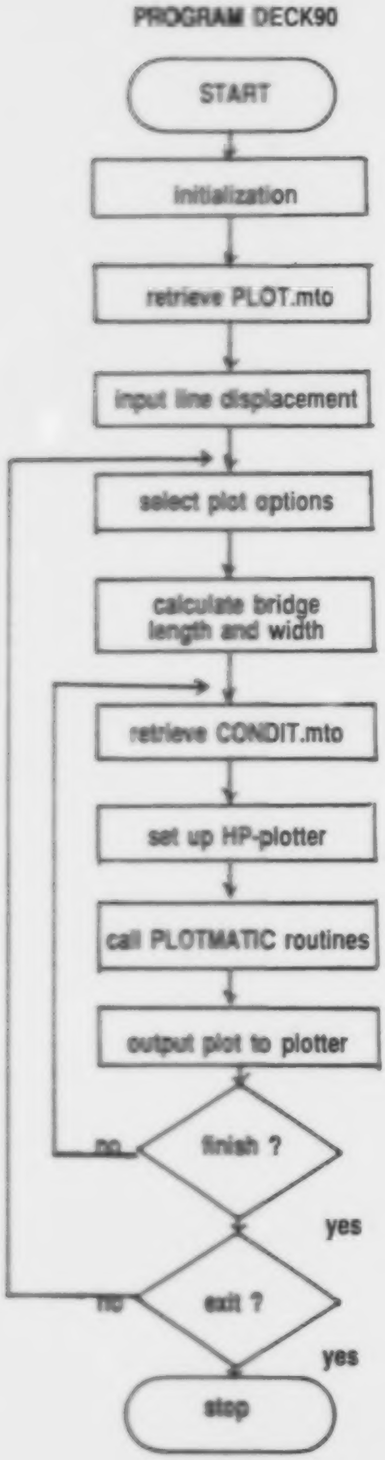

PROGRAM DECK90
START
initialization
retrieve PLOT.mto
input line displacement
select plot options
calculate bridge length and width
retrieve CONDIT.mto
set up HP-plotter
call PLOTMATIC routines
output plot to plotter
no
finish ?
yes
no
exit ?
yes
stop

Figure 11: Flow Diagram for Program DECK90

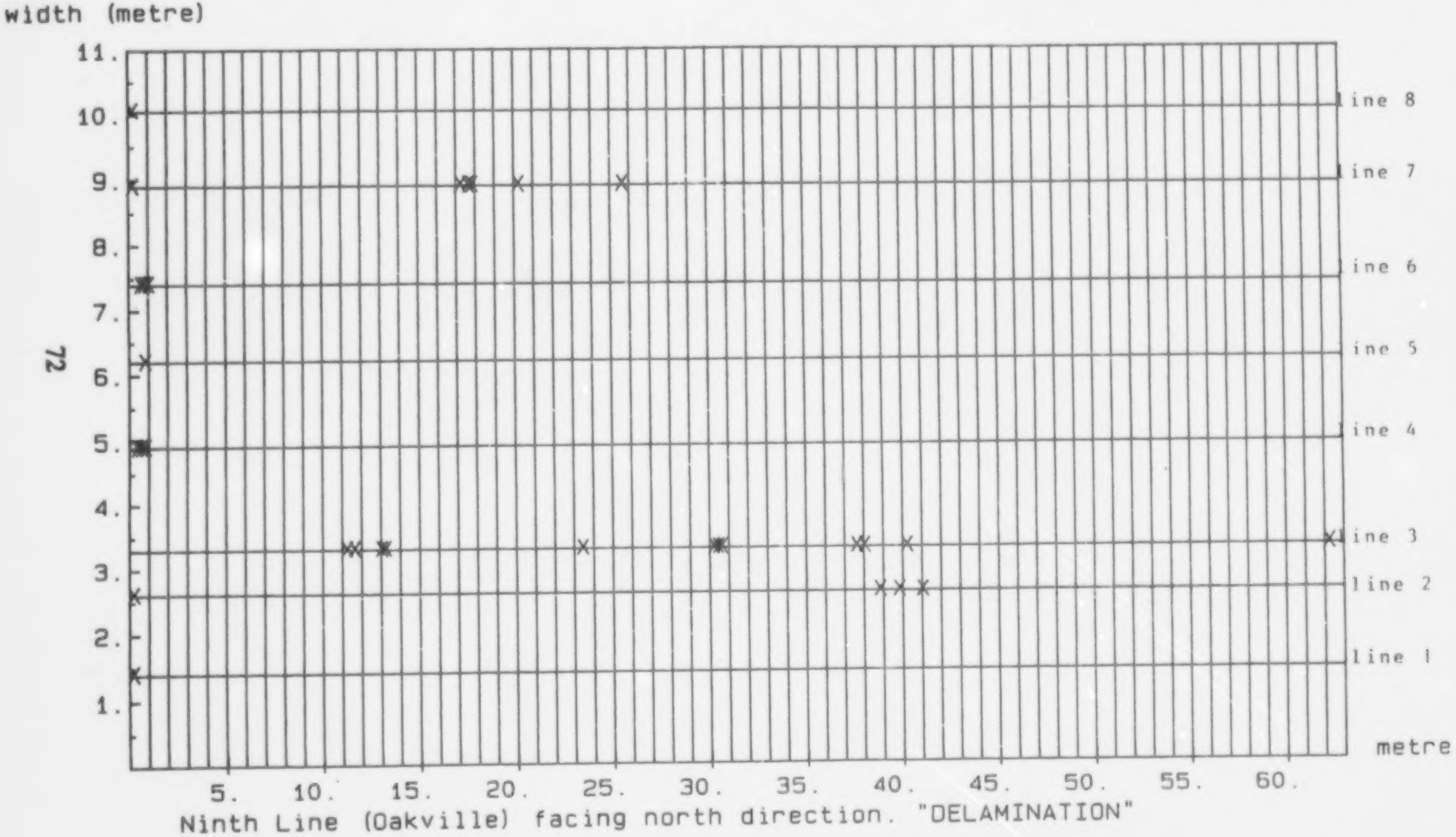


Figure 12: Delaminated Areas at Ninth Line Detected by XT-DART (1988).

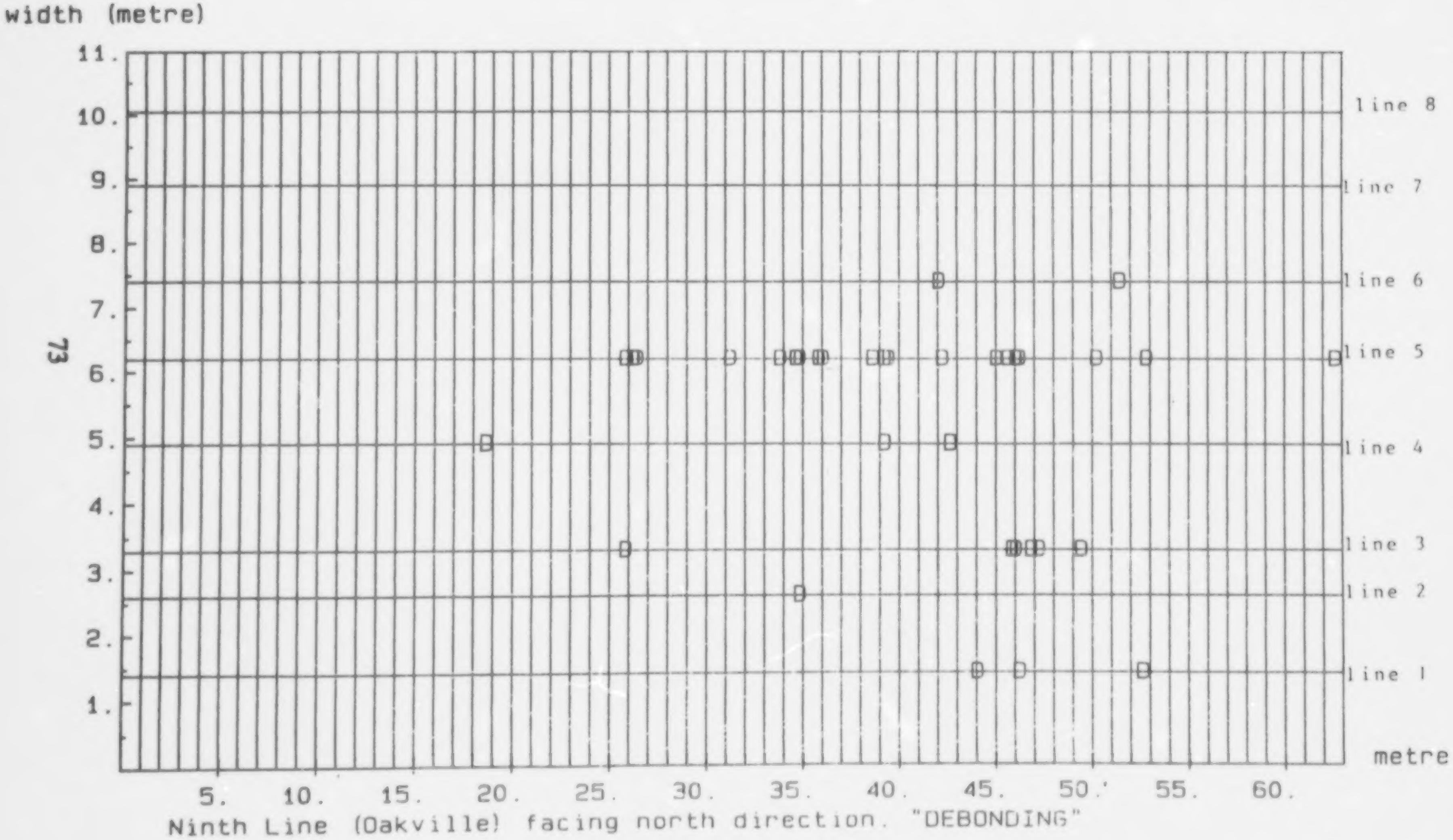


Figure 13: Debonded Areas at Ninth Line Detected by XT-DART (1988).

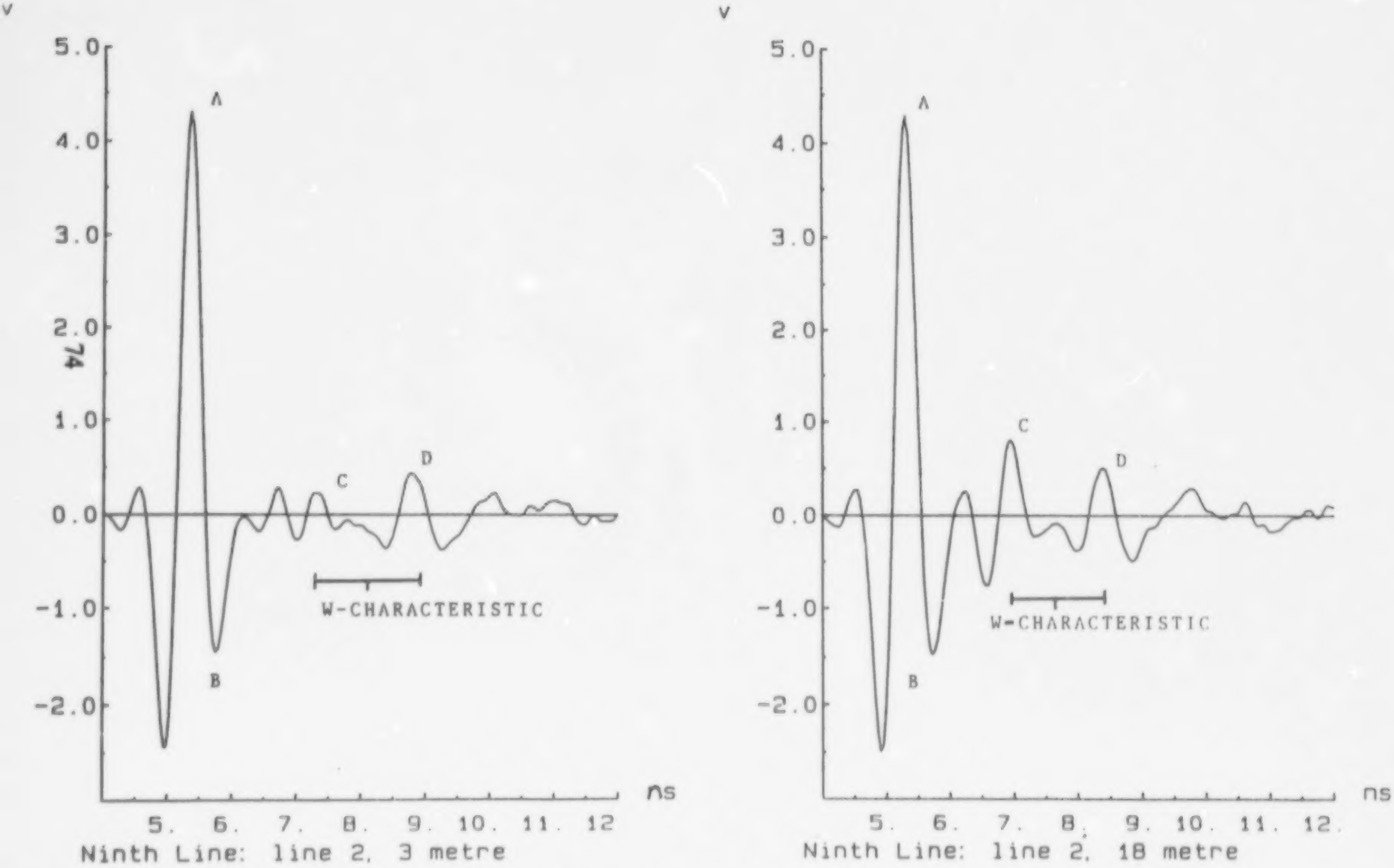
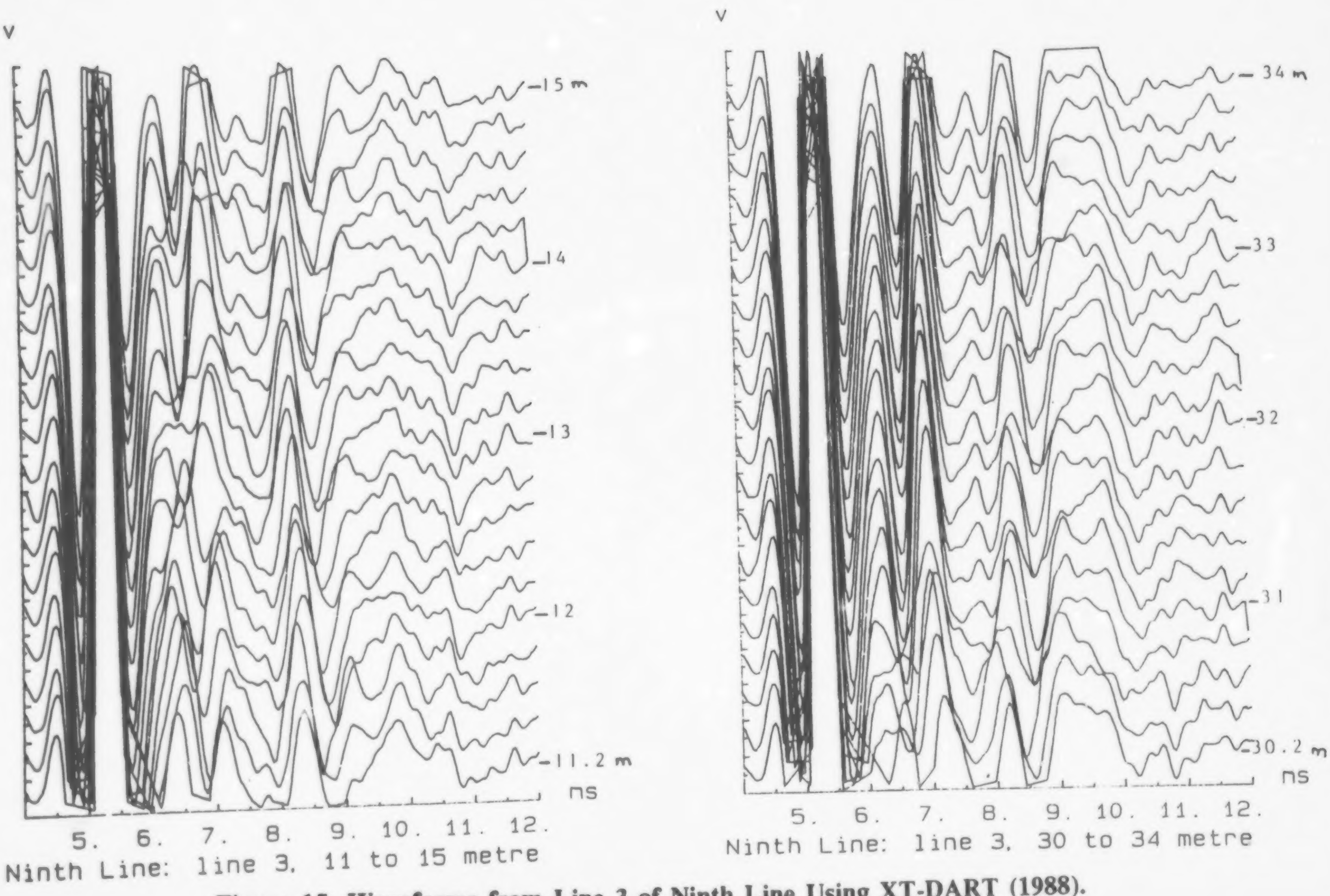


Figure 14: Sample Waveforms from Ninth Line Using XT-DART (1988).



Ninth Line: line 3, 11 to 15 metre

Ninth Line: line 3, 30 to 34 metre

Figure 15: Waveforms from Line 3 of Ninth Line Using XT-DART (1988).

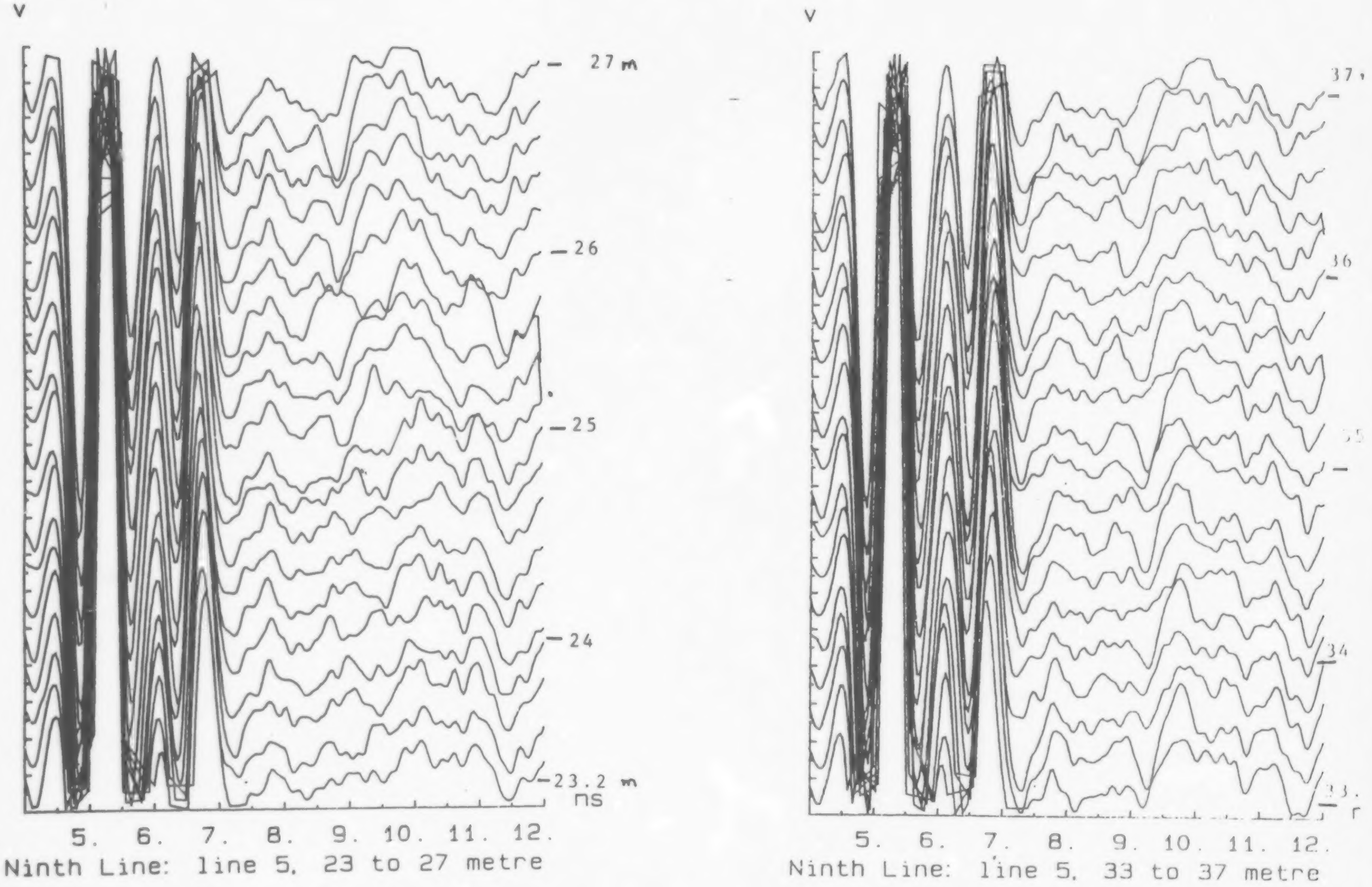
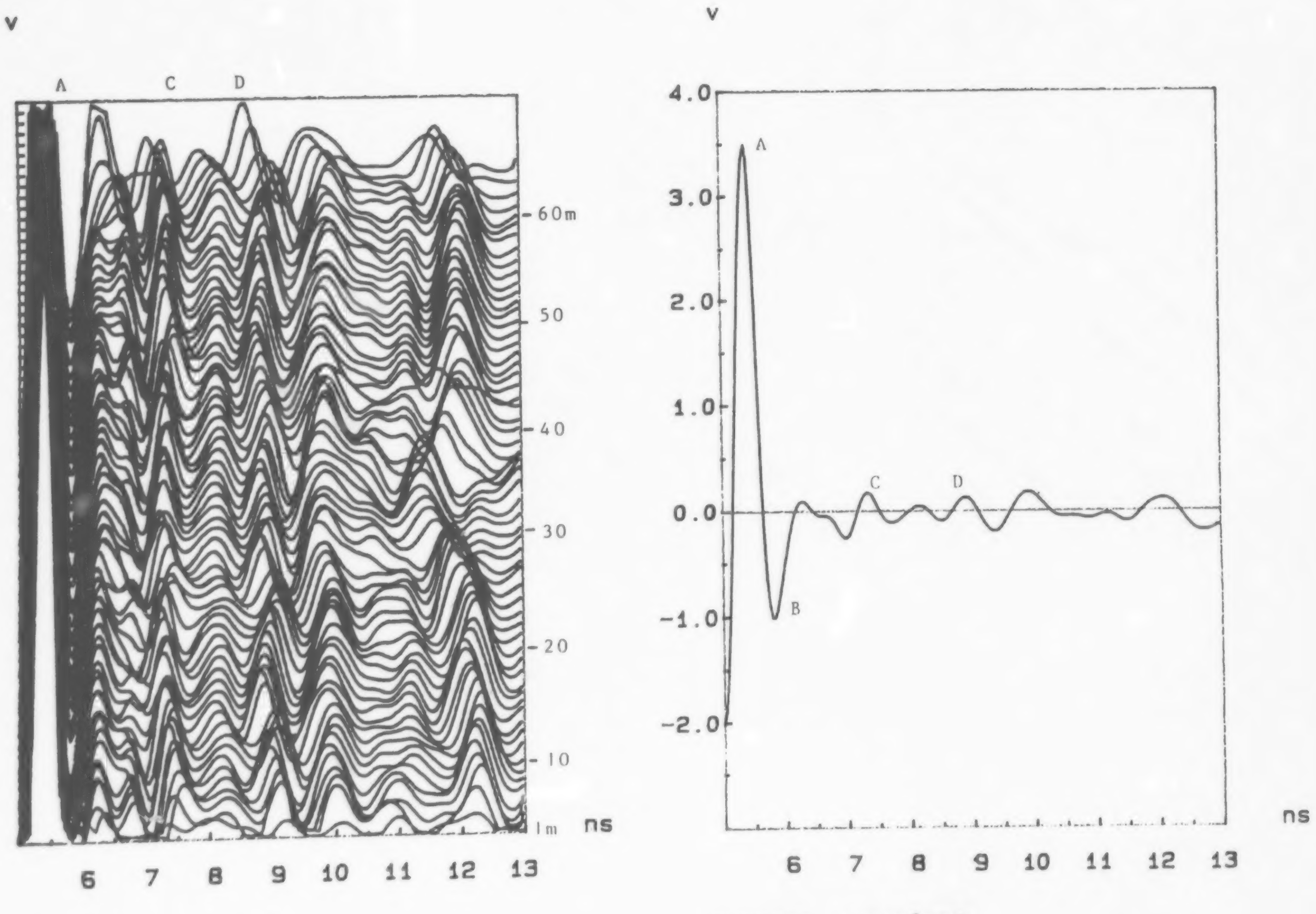


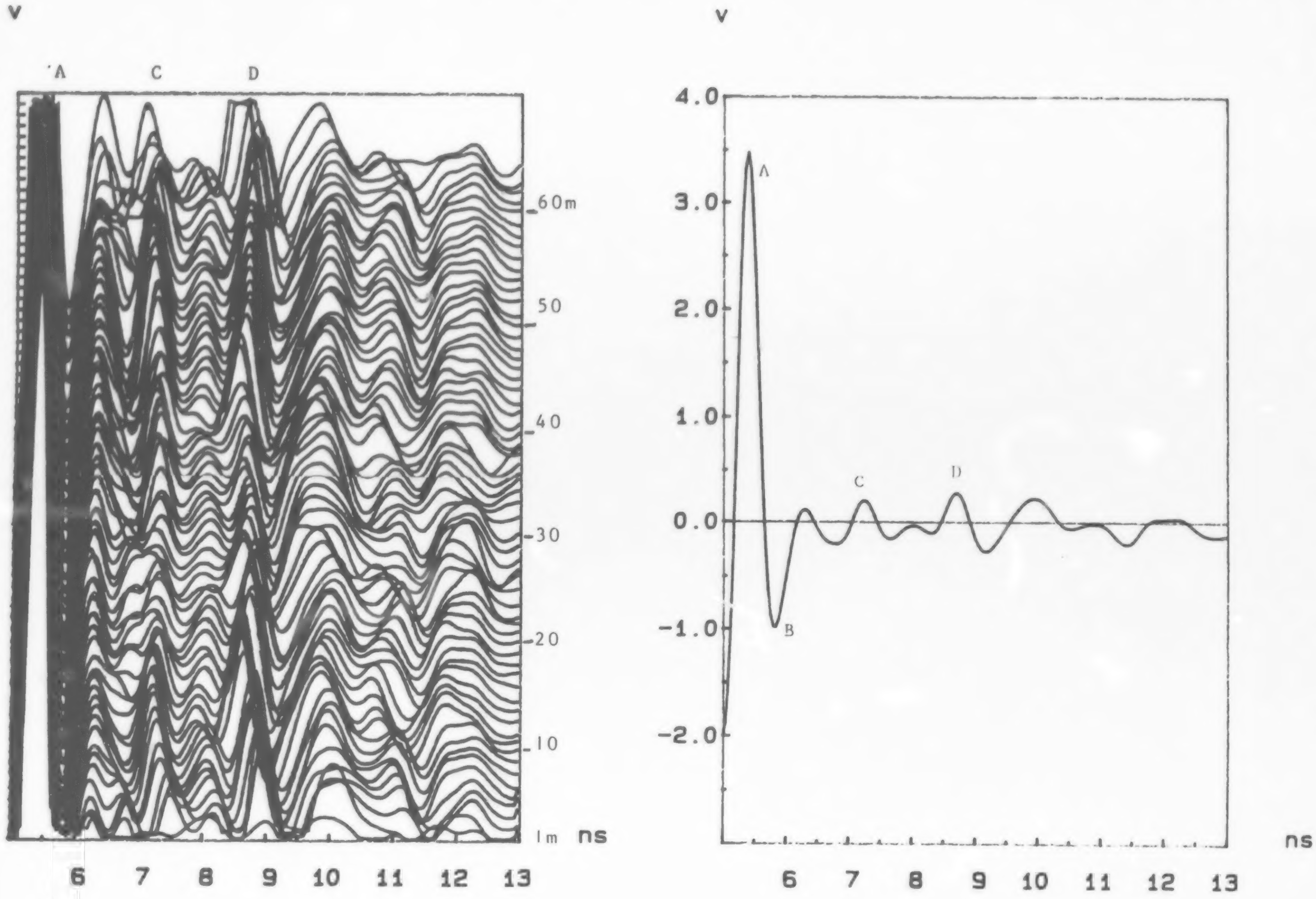
Figure 16: Waveforms from Line 5 of Ninth Line Using XT-DART (1988).



Ninth Line. line 1

average waveform

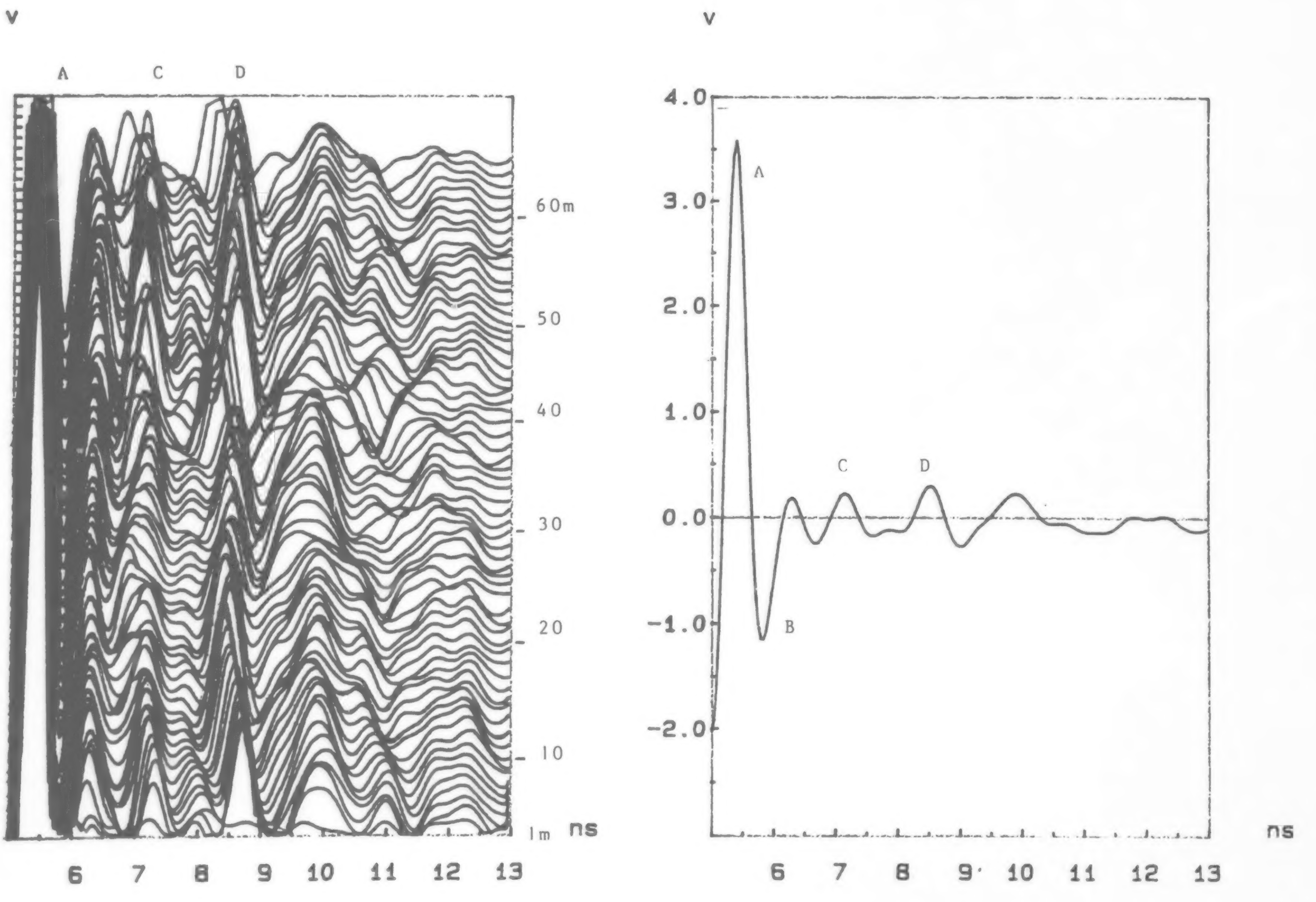
Figure 17: Waveforms from Line 1 of Ninth Line Sampled by 386-DART.



Ninth Line, line 2

average waveform

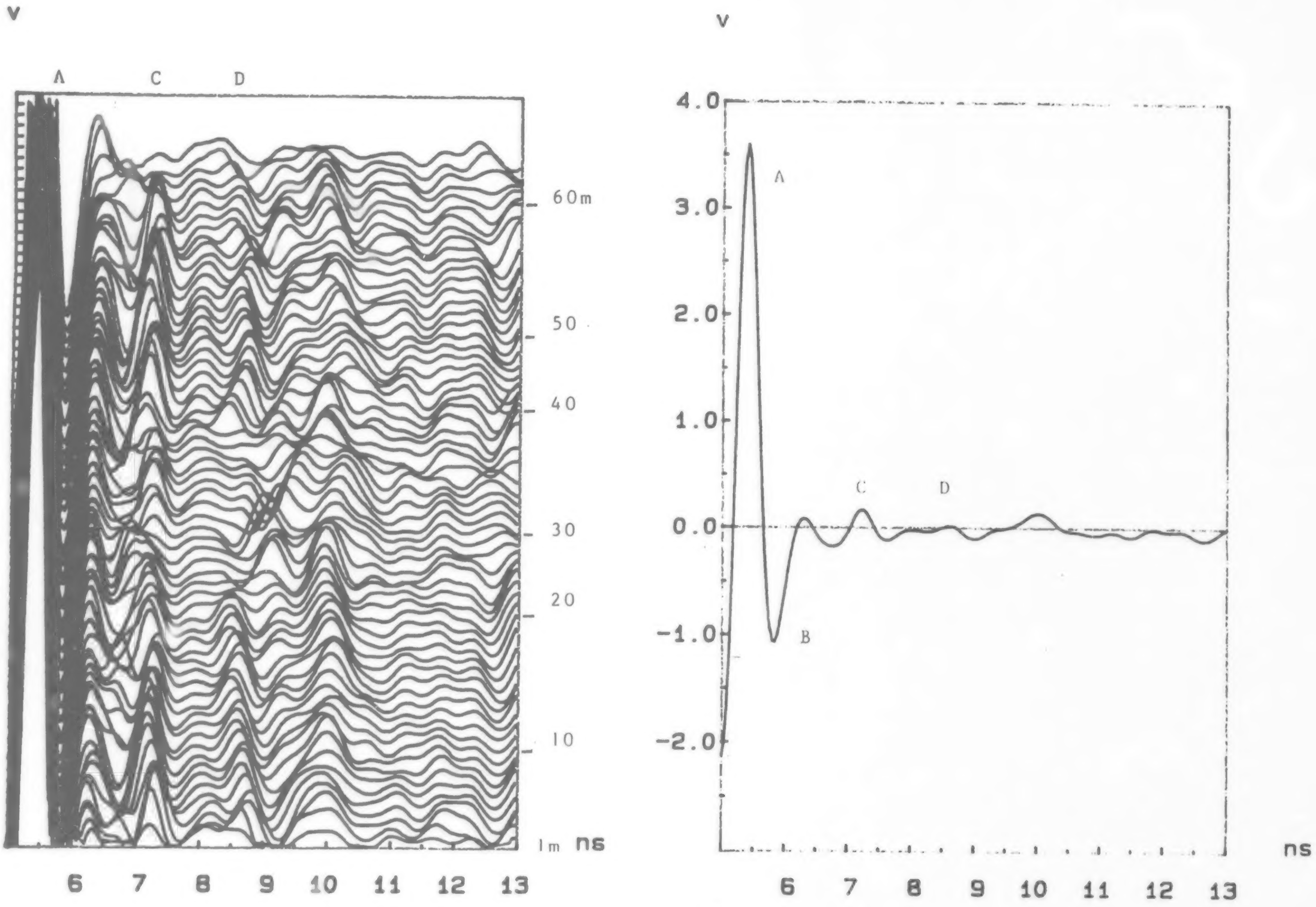
Figure 18: Waveforms from Line 2 of Ninth Line Sampled by 386-DART.



Ninth Line, line 3

average waveform

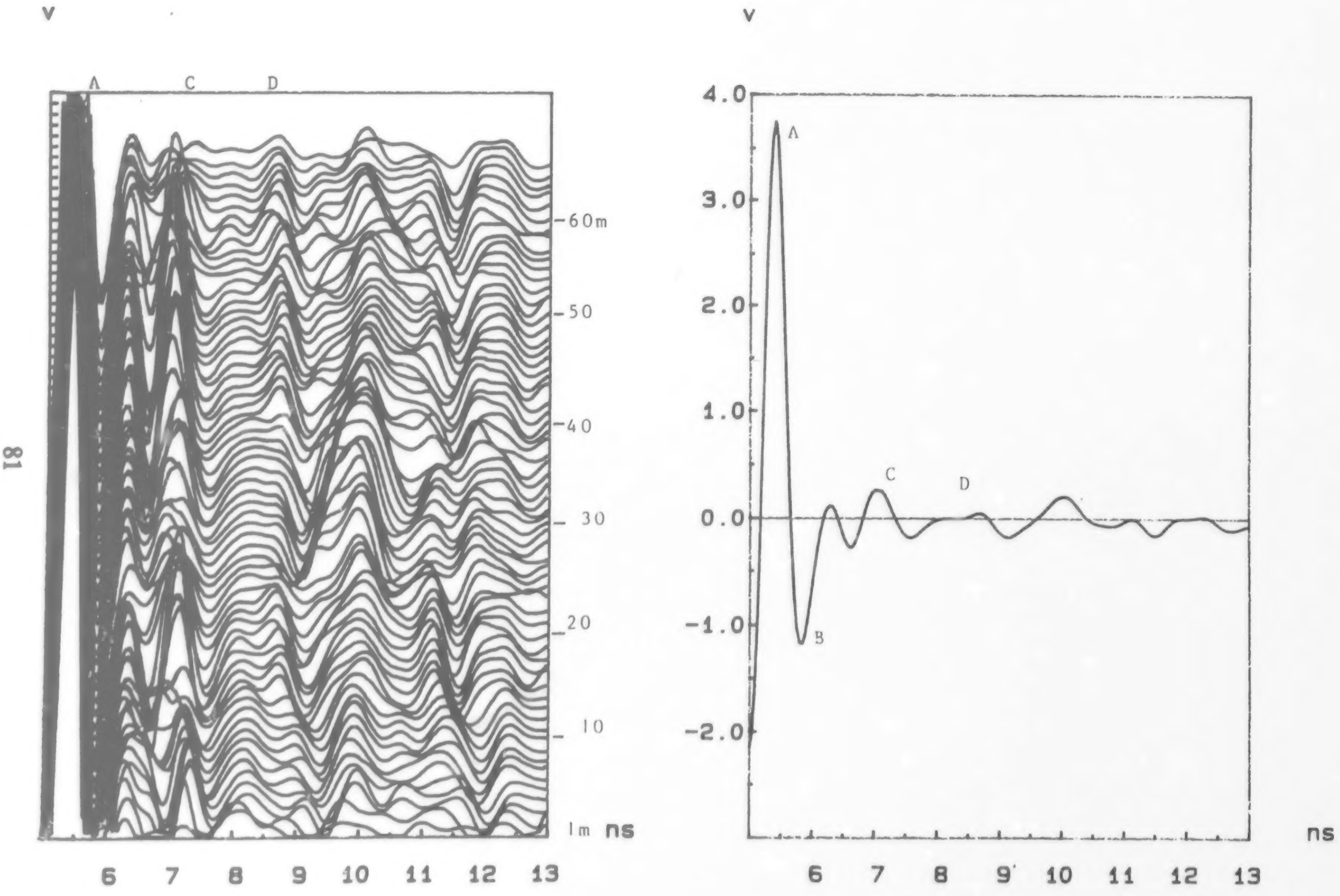
Figure 19: Waveforms from Line 3 of Ninth Line Sampled by 386-DART.



Ninth Line, line 4

average waveform

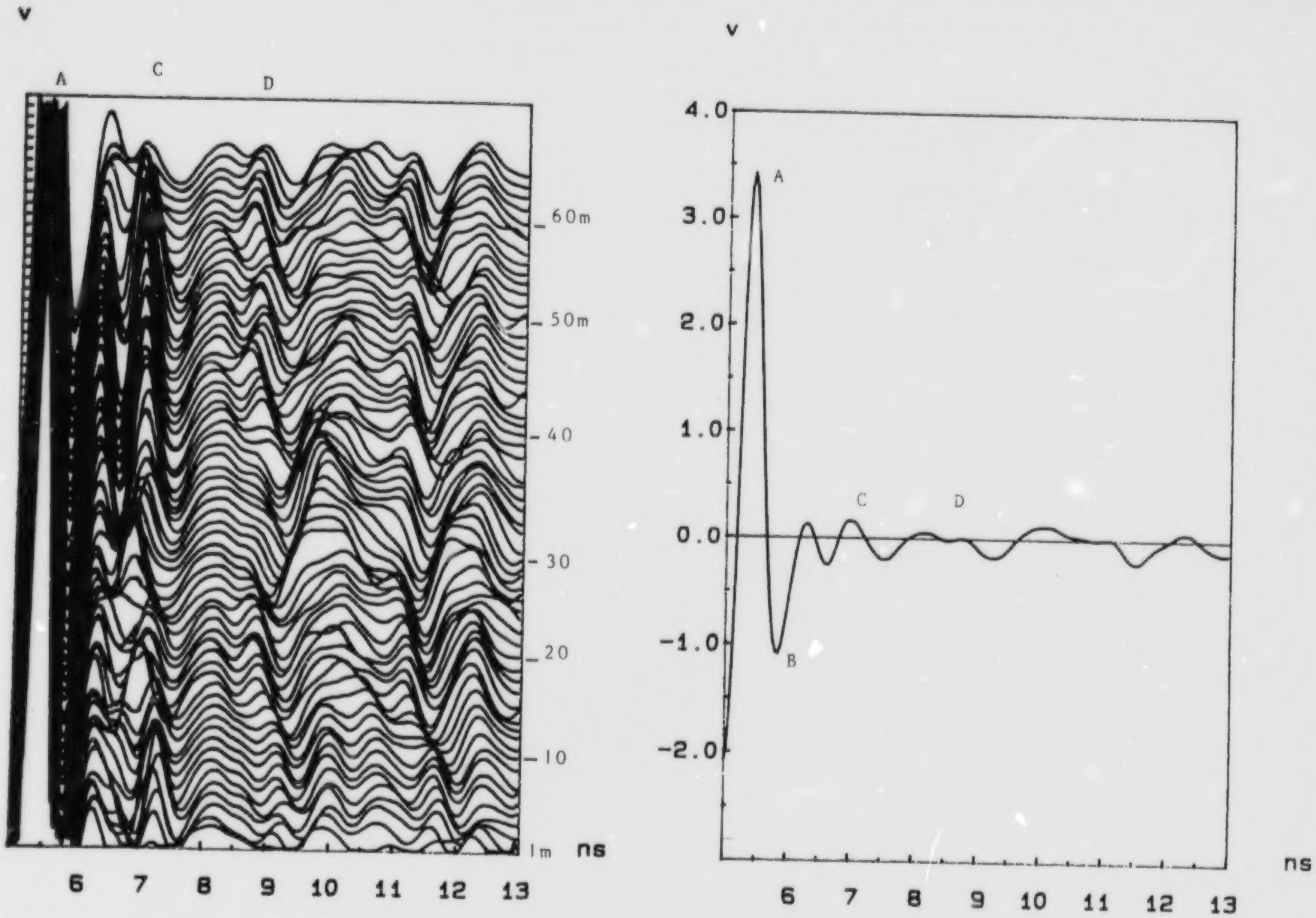
Figure 20: Waveforms from Line 4 of Ninth Line Sampled by 386-DART.



Ninth Line. line 5

average waveform

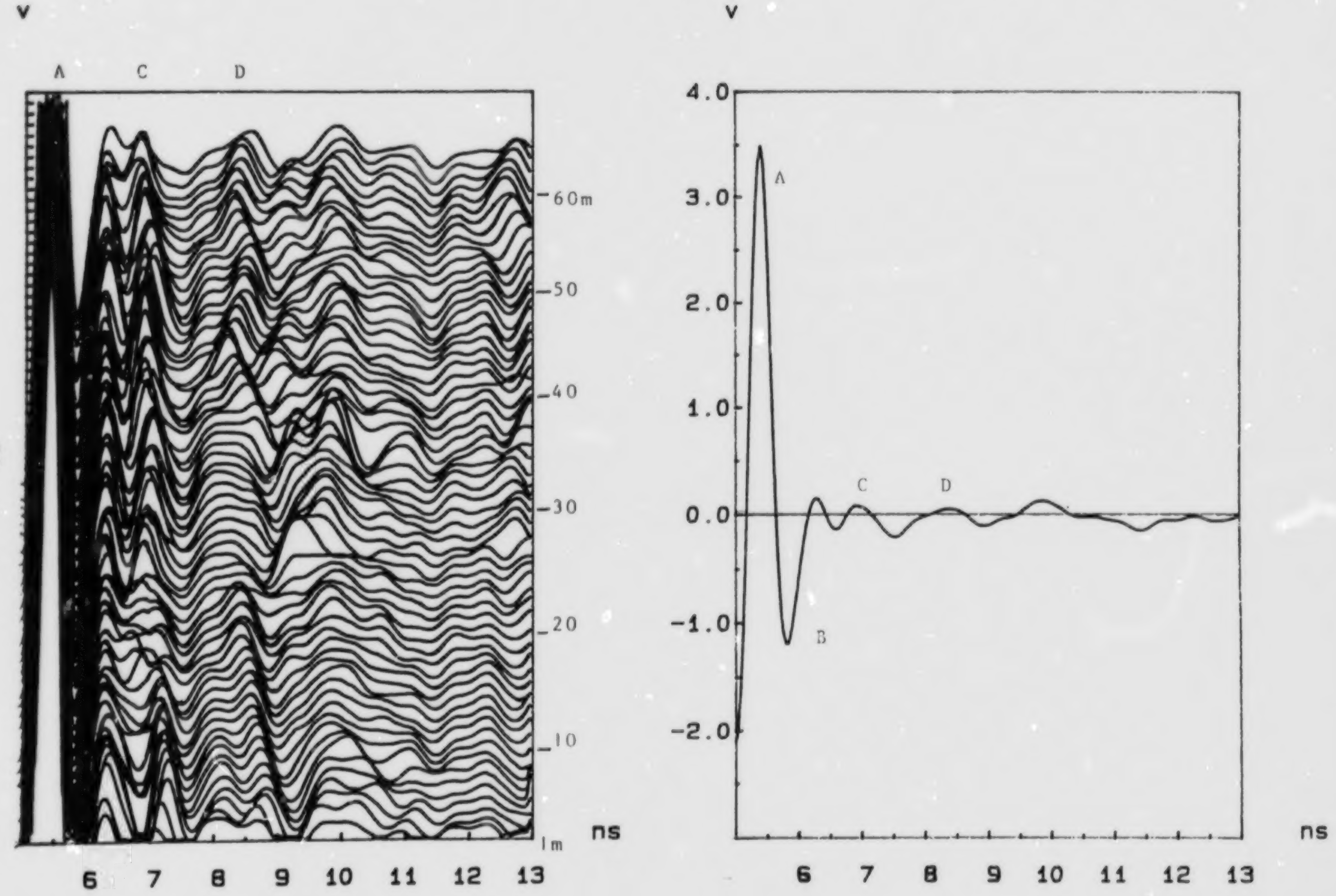
Figure 21: Waveforms from Line 5 of Ninth Line Sampled by 386-DART.



Ninth Line, line 6

average waveform

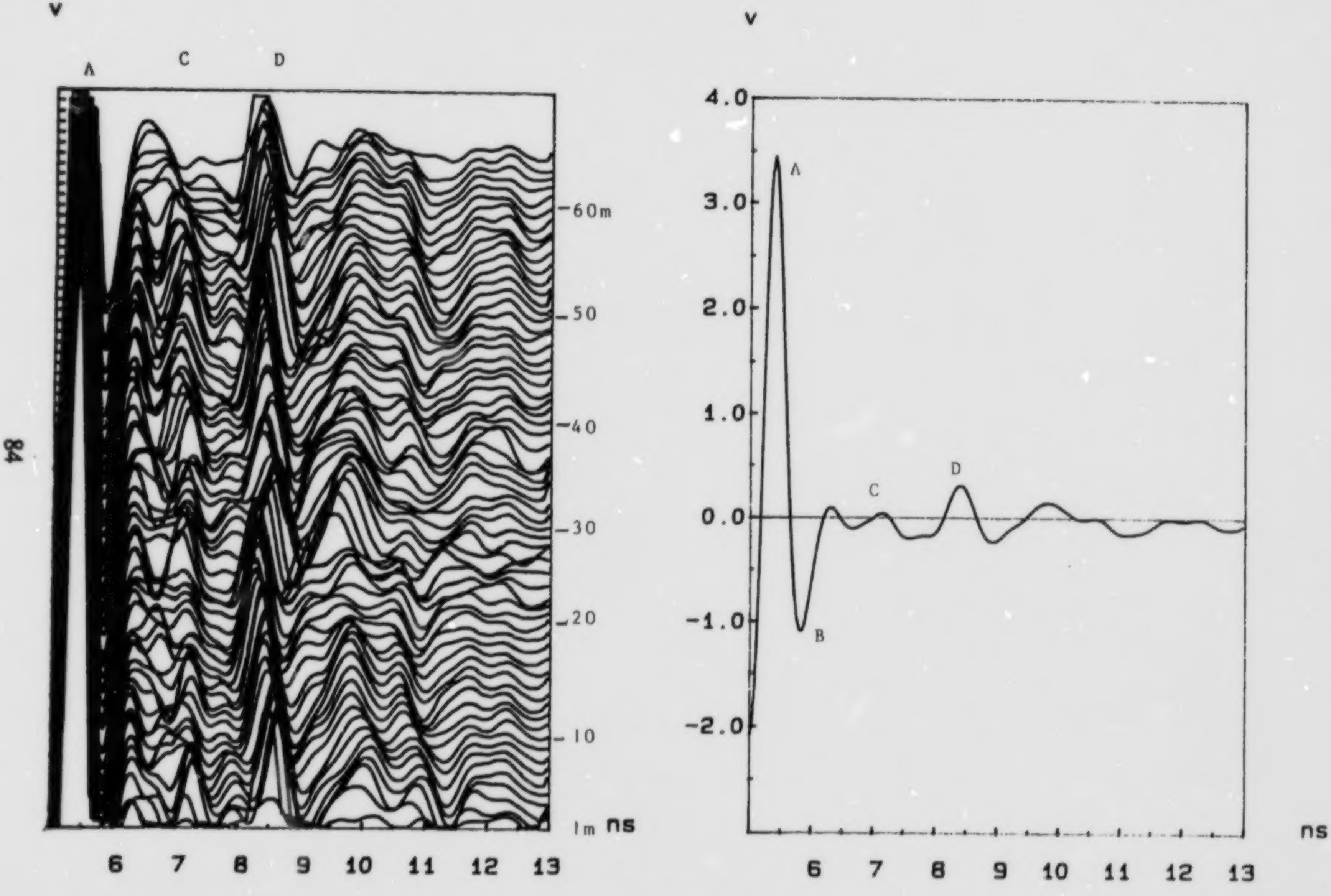
Figure 22: Waveforms from Line 6 of Ninth Line Sampled by 386-DART.



Ninth Line. line 7

average waveform

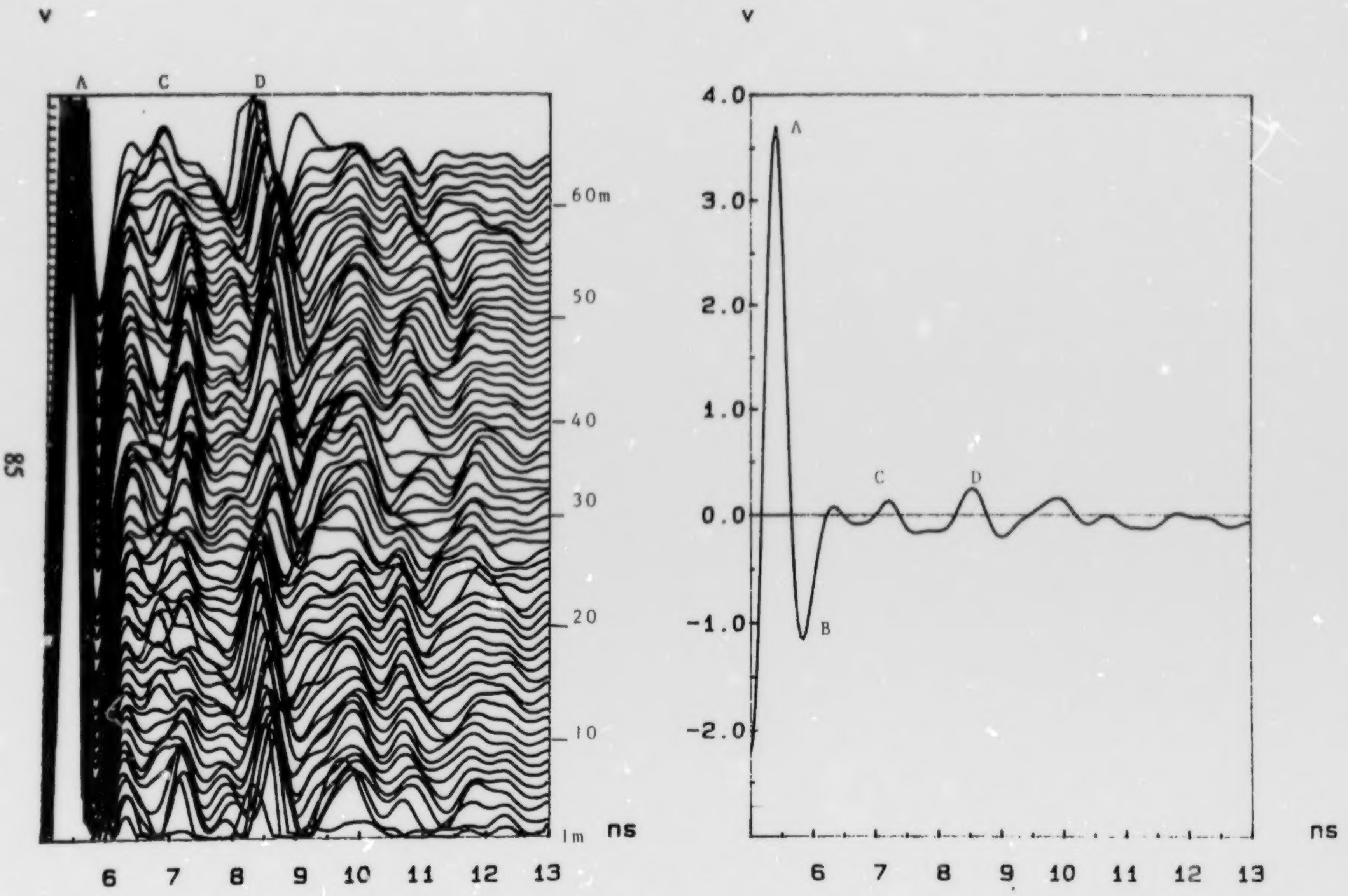
Figure 23: Waveforms from Line 7 of Ninth Line Sampled by 386-DART.



Ninth Line, line 8

average waveform

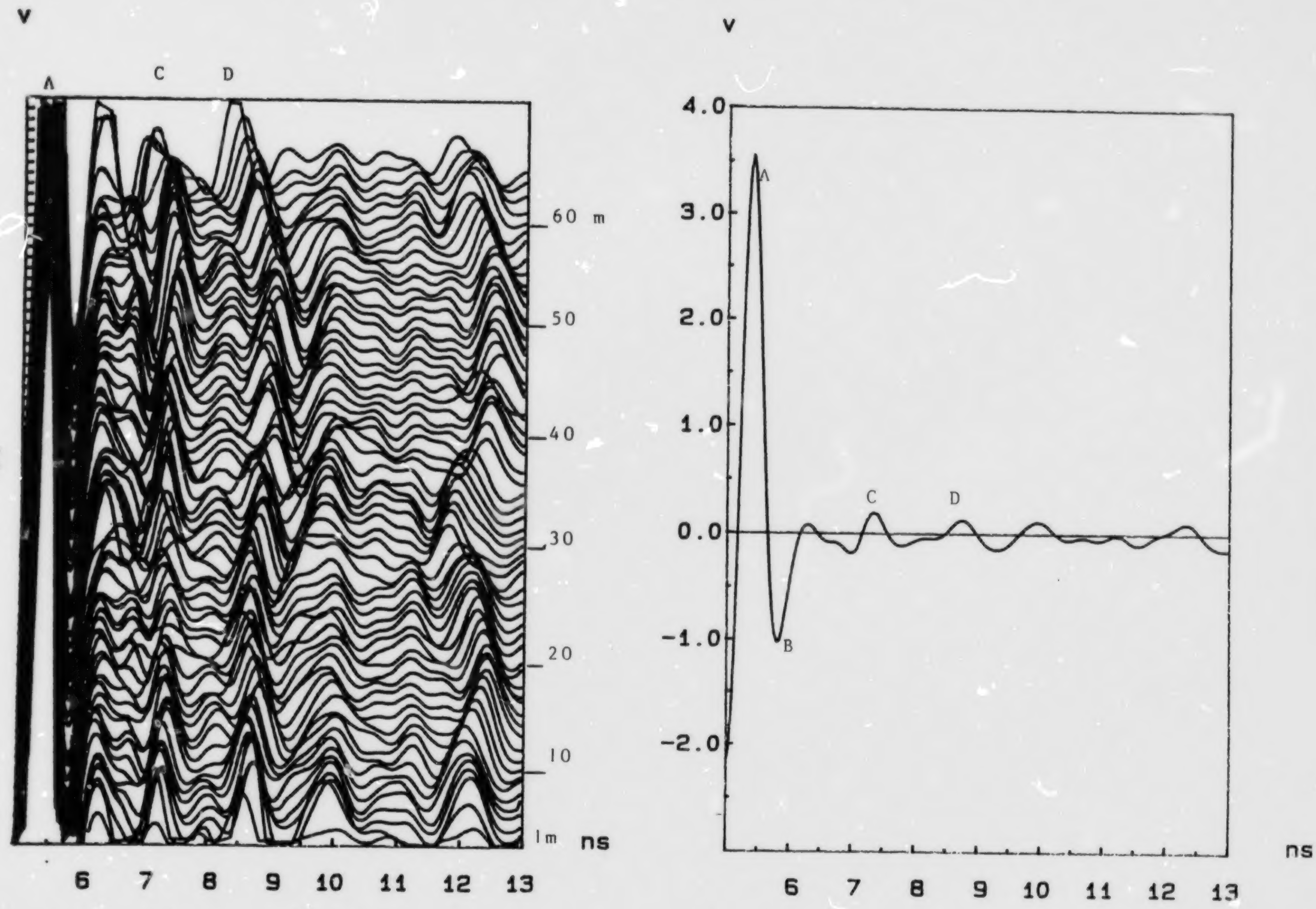
Figure 24: Waveforms from Line 8 of Ninth Line Sampled by 386-DART.



Ninth Line. line 9

average waveform

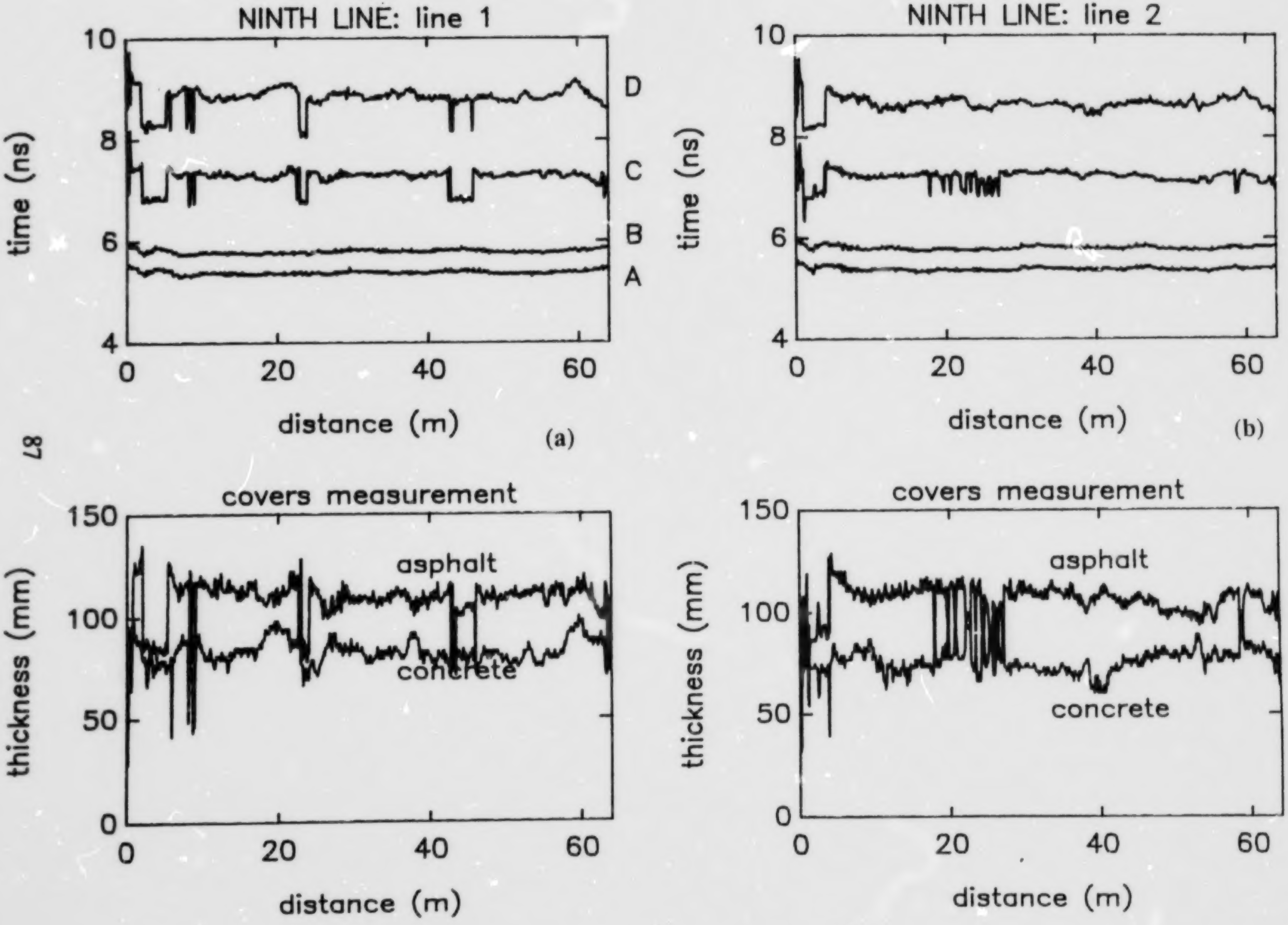
Figure 25: Waveforms from Line 9 of Ninth Line Sampled by 386-DART.



Ninth Line. line 10

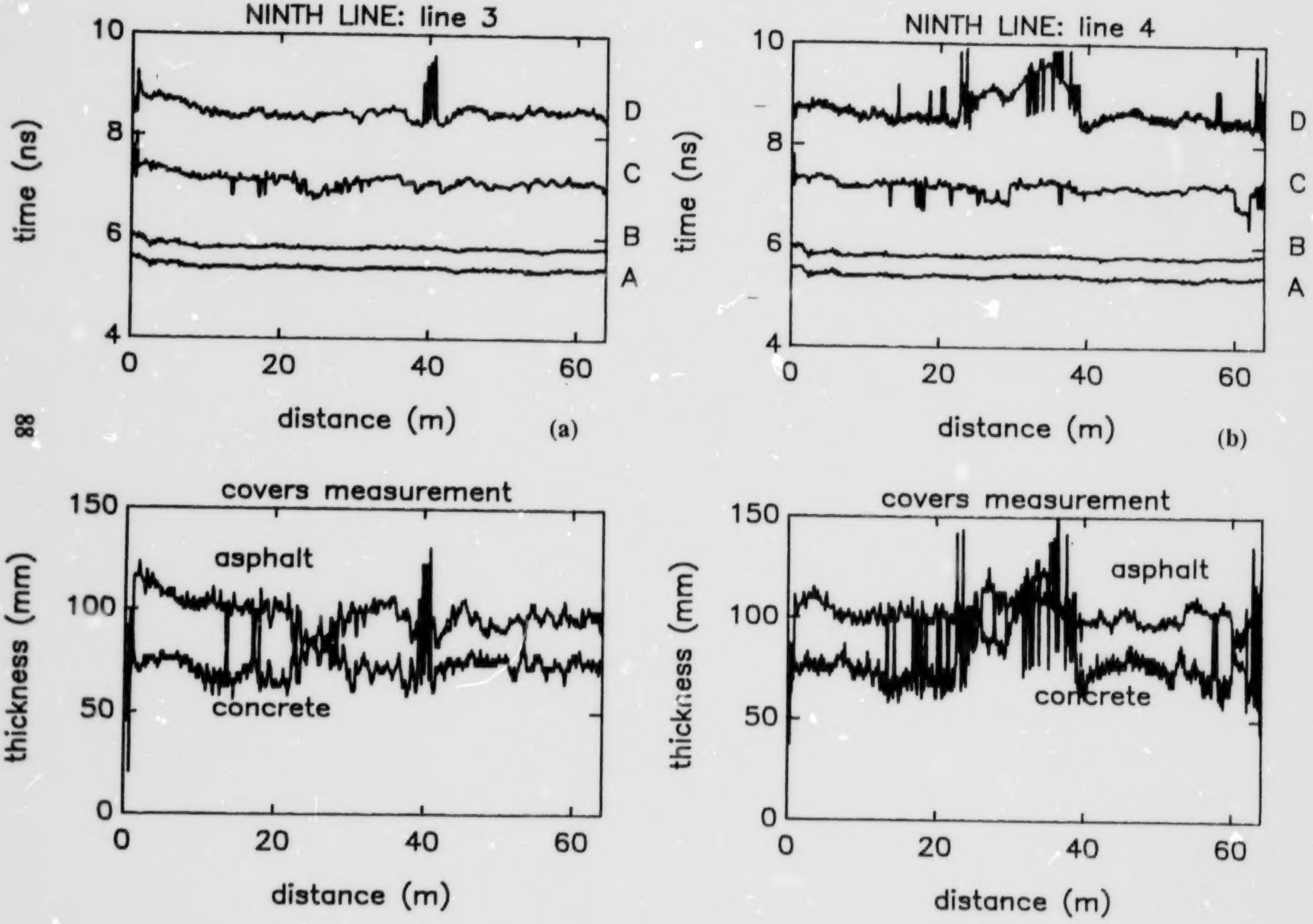
average waveform

Figure 26: Waveforms from Line 10 of Ninth Line Sampled by 386-DART.



(a) (b)

Figure 27: Ninth Line: Peaks and Covers Measurement (a) Line 1 (b) Line 2.



(a)

(b)

Figure 28: Ninth Line: Peaks and Covers Measurement (a) Line 3 (b) Line 4.

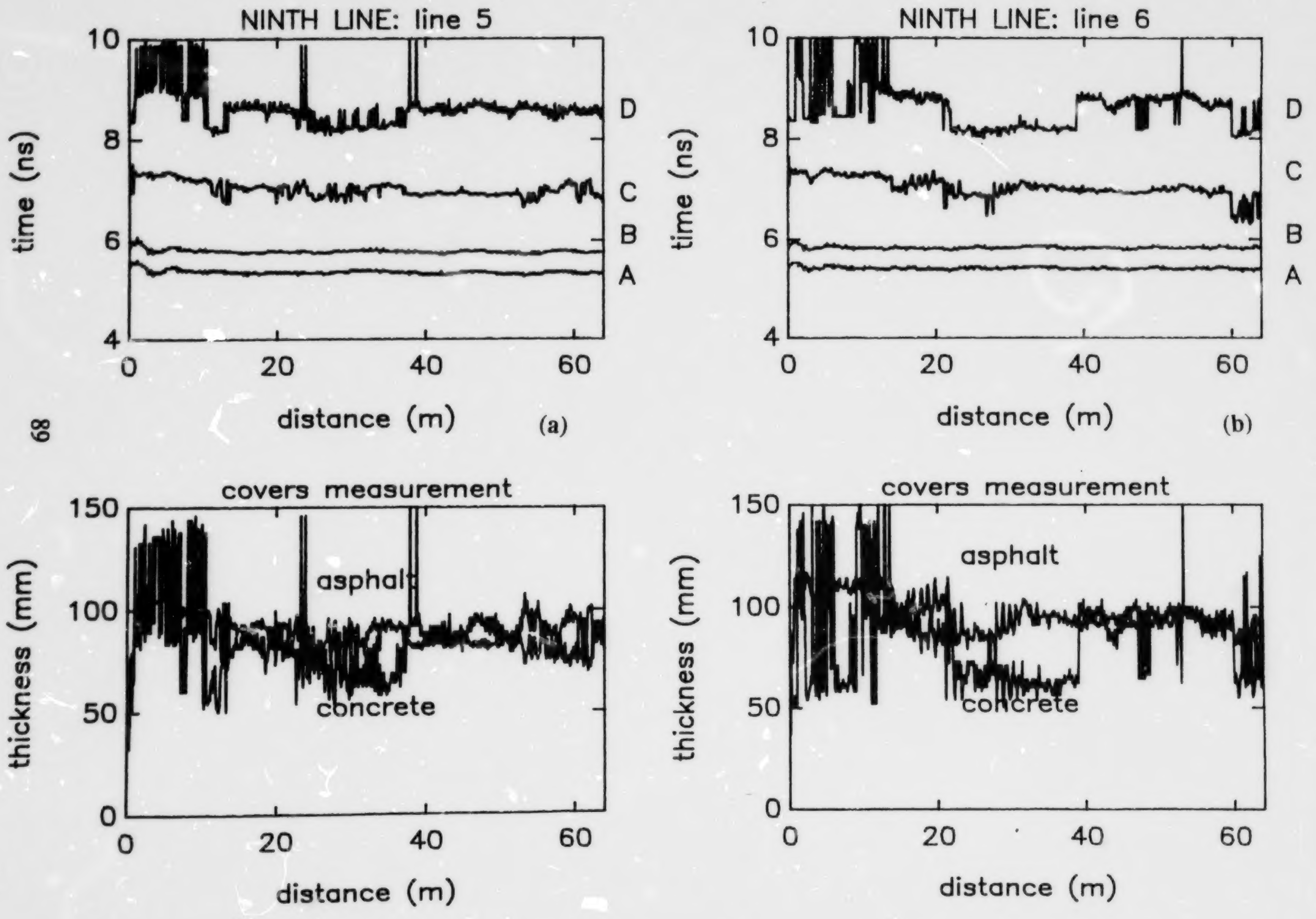


Figure 29: Ninth Line: Peaks and Covers Measurement (a) Line 5 (b) Line 6.

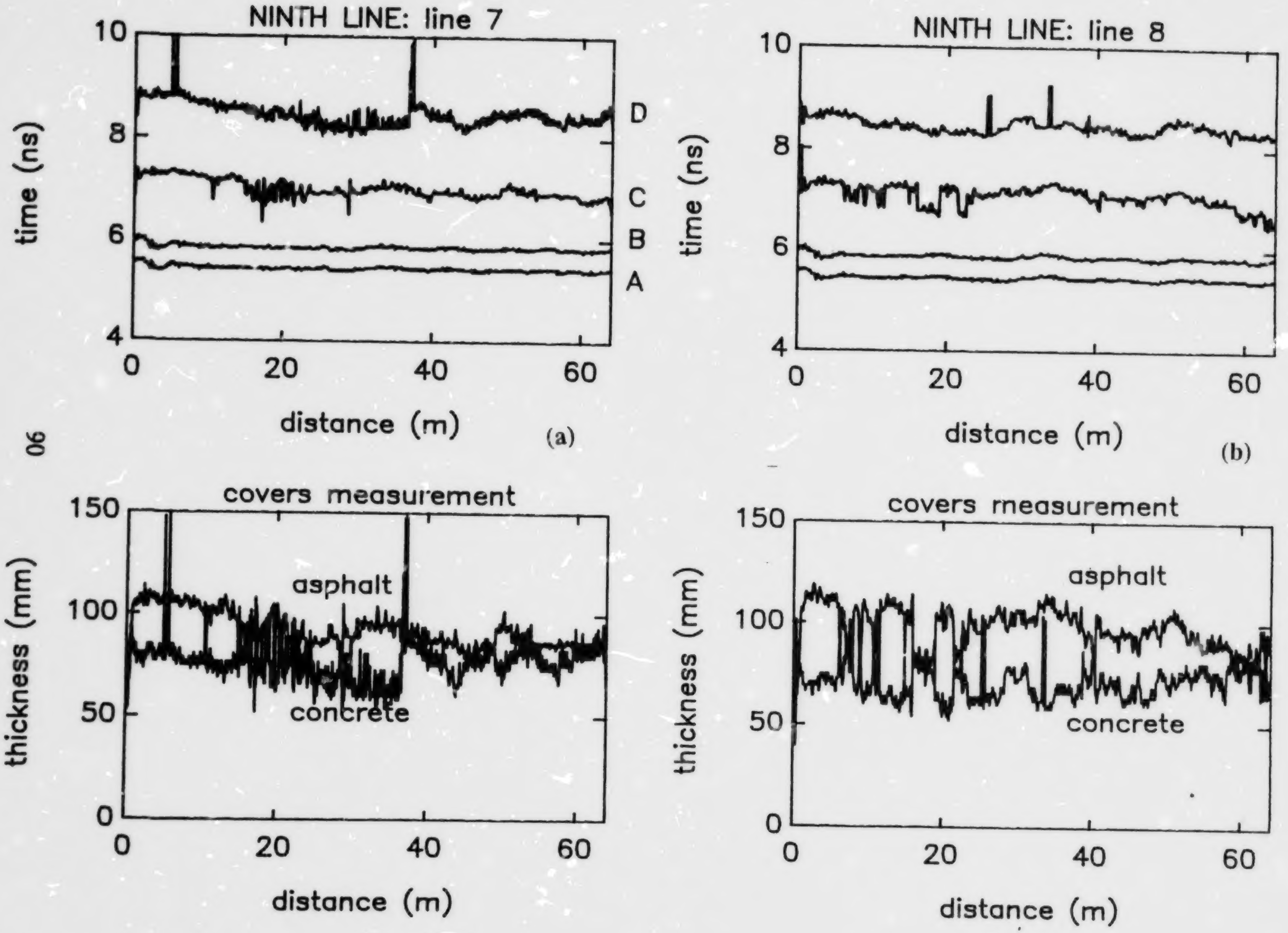


Figure 30: Ninth Line: Peaks and Covers Measurement (a) Line 7 (b) Line 8.

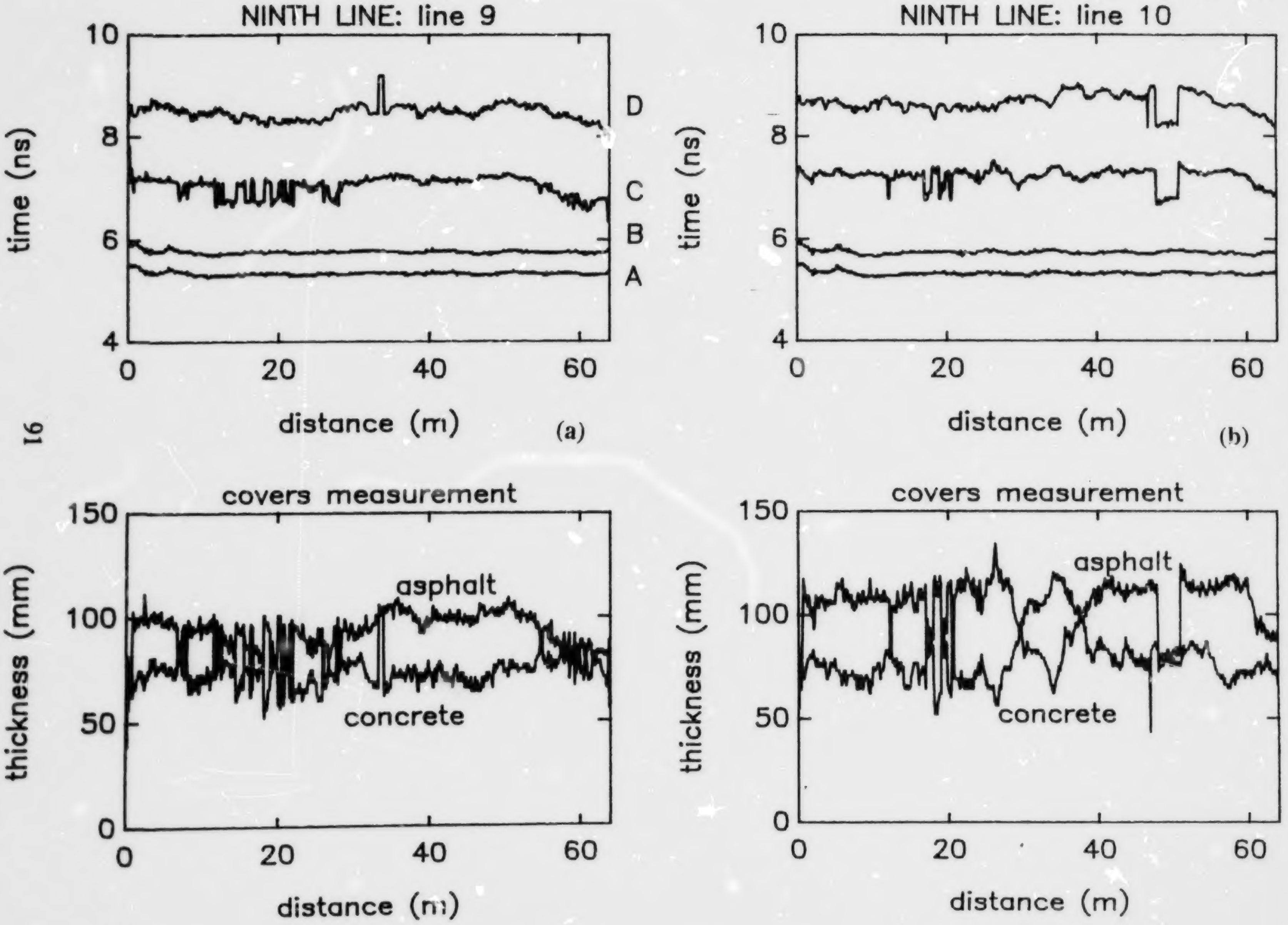


Figure 31: Ninth Line: Peaks and Covers Measurement (a) Line 9 (b) Line 10

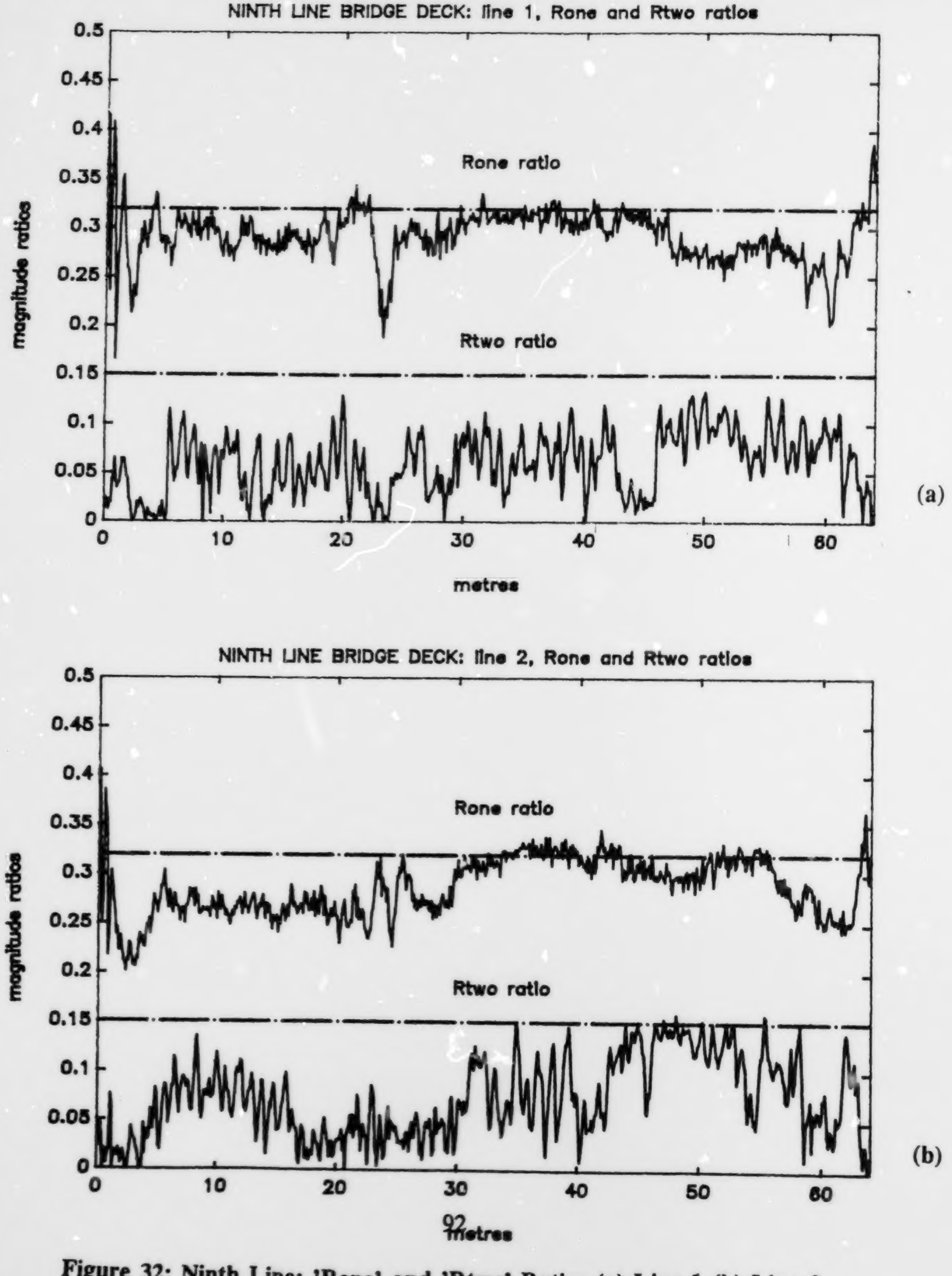


Figure 32: Ninth Line: 'Rone' and 'Rtwo' Ratios (a) Line 1 (b) Line 2

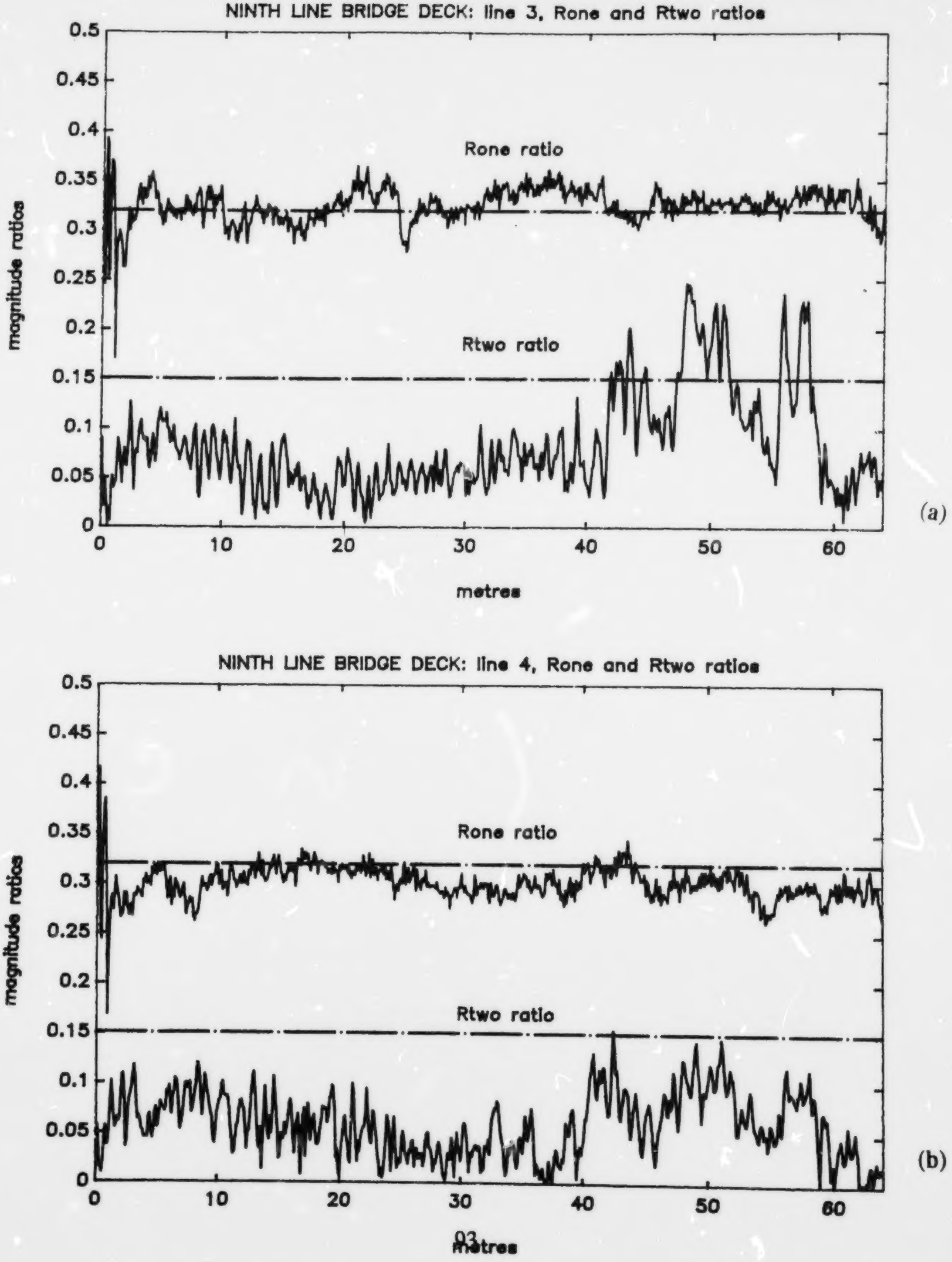
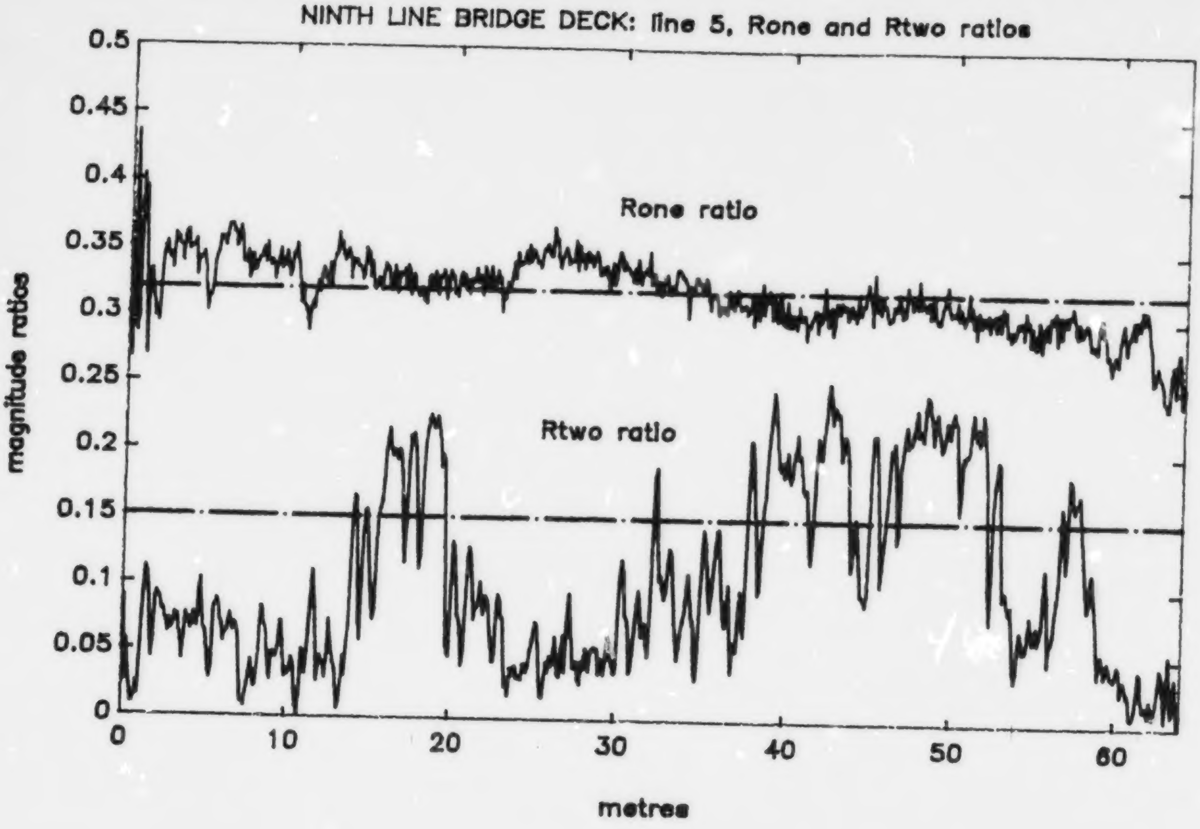
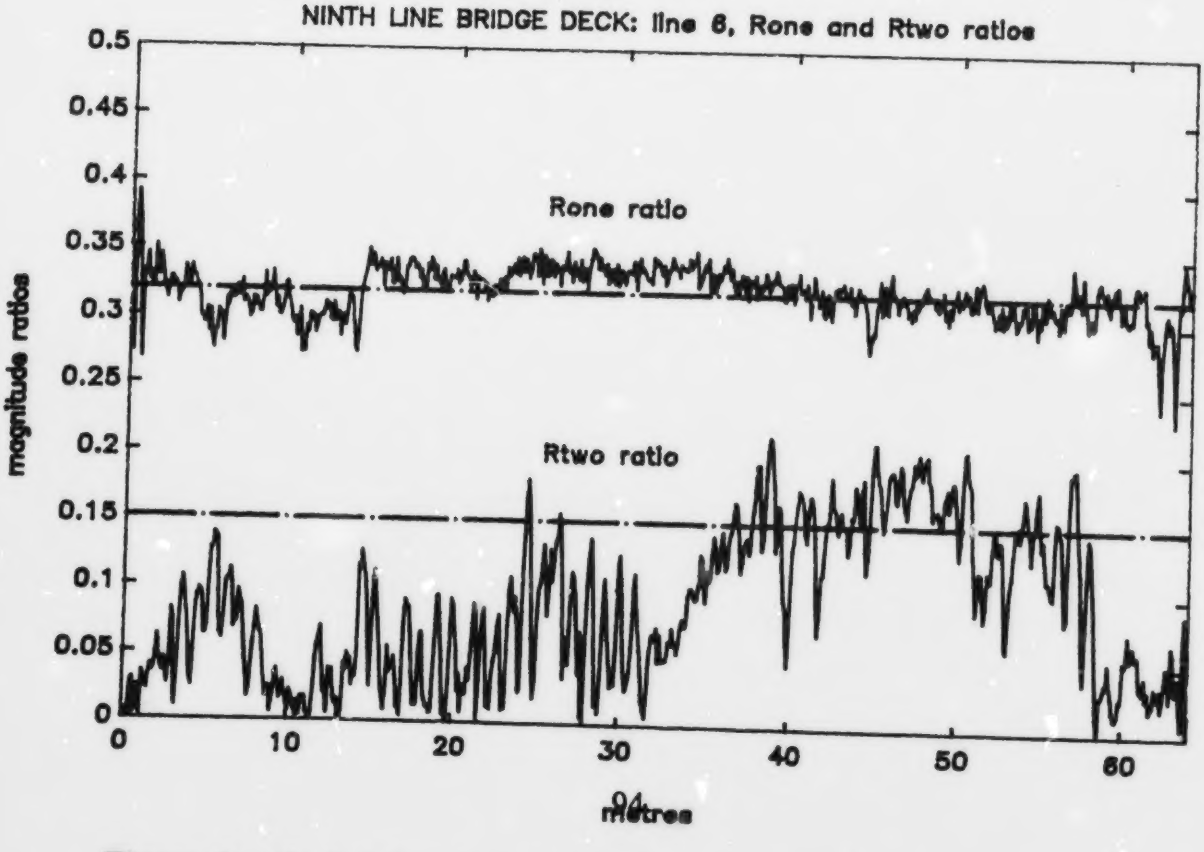


Figure 33: Ninth Line: 'Rone' and 'Rtwo' Ratios (a) Line 3 (b) Line 4.



(a)



(b)

Figure 34: Ninth Line: 'Rone' and 'Rtwo' Ratios (a) Line 5 (b) Line 6

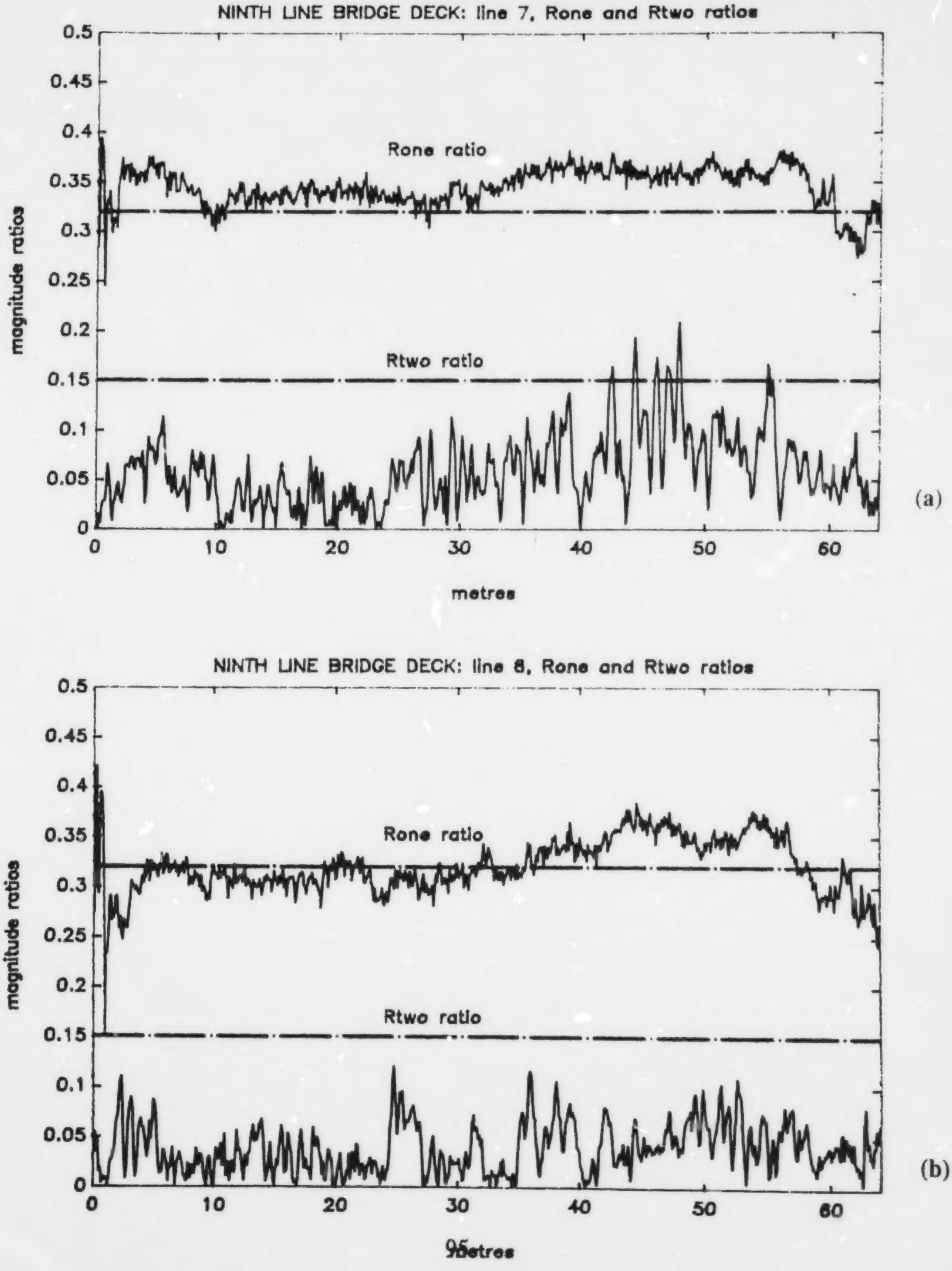


Figure 35: Ninth Line: 'Rone' and 'Rtwo' Ratios (a) Line 7 (b) Line 8

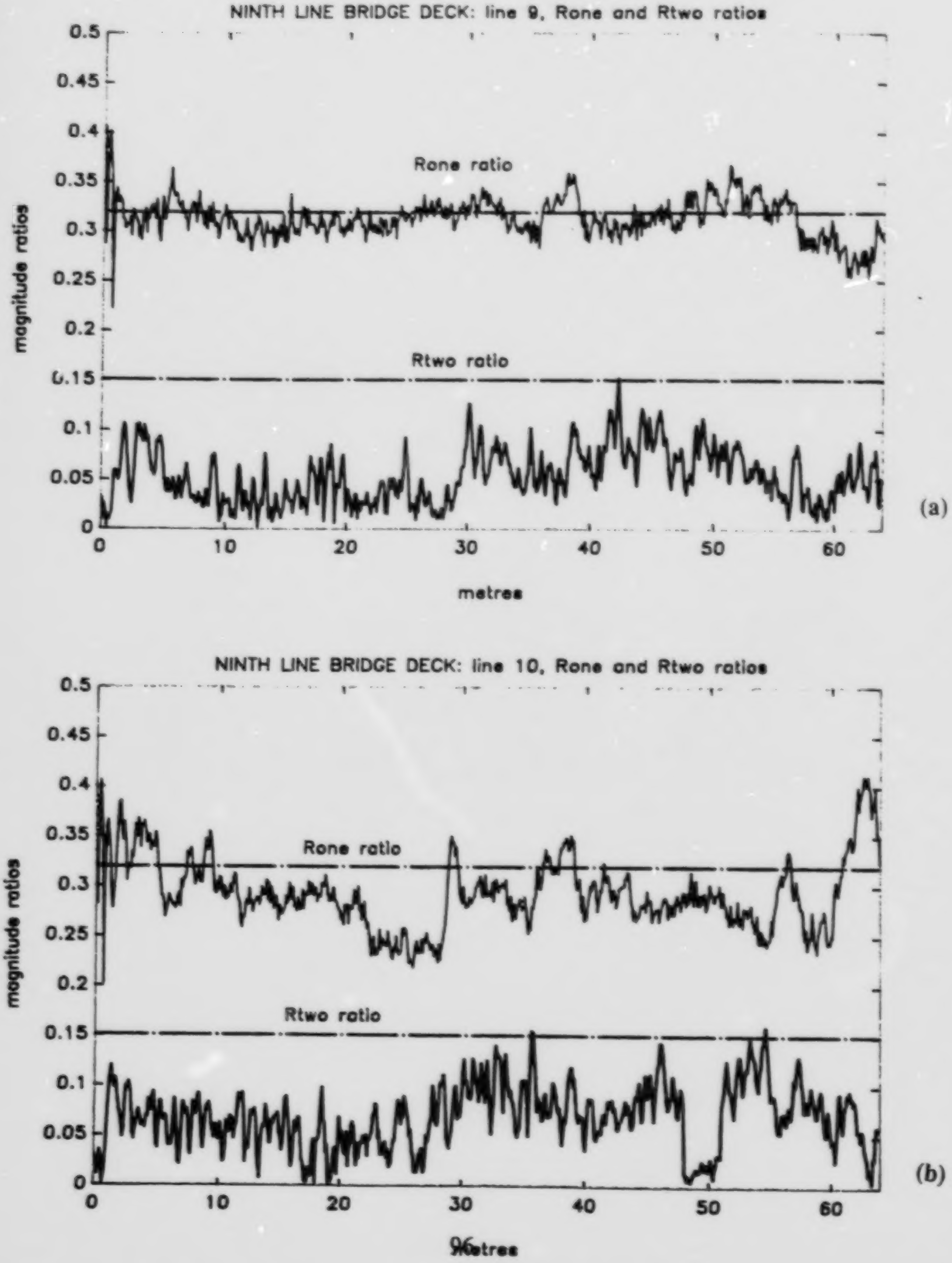


Figure 36: Ninth Line: 'Rone' and 'Rtwo' Ratios (a) Line 9 (b) Line 10

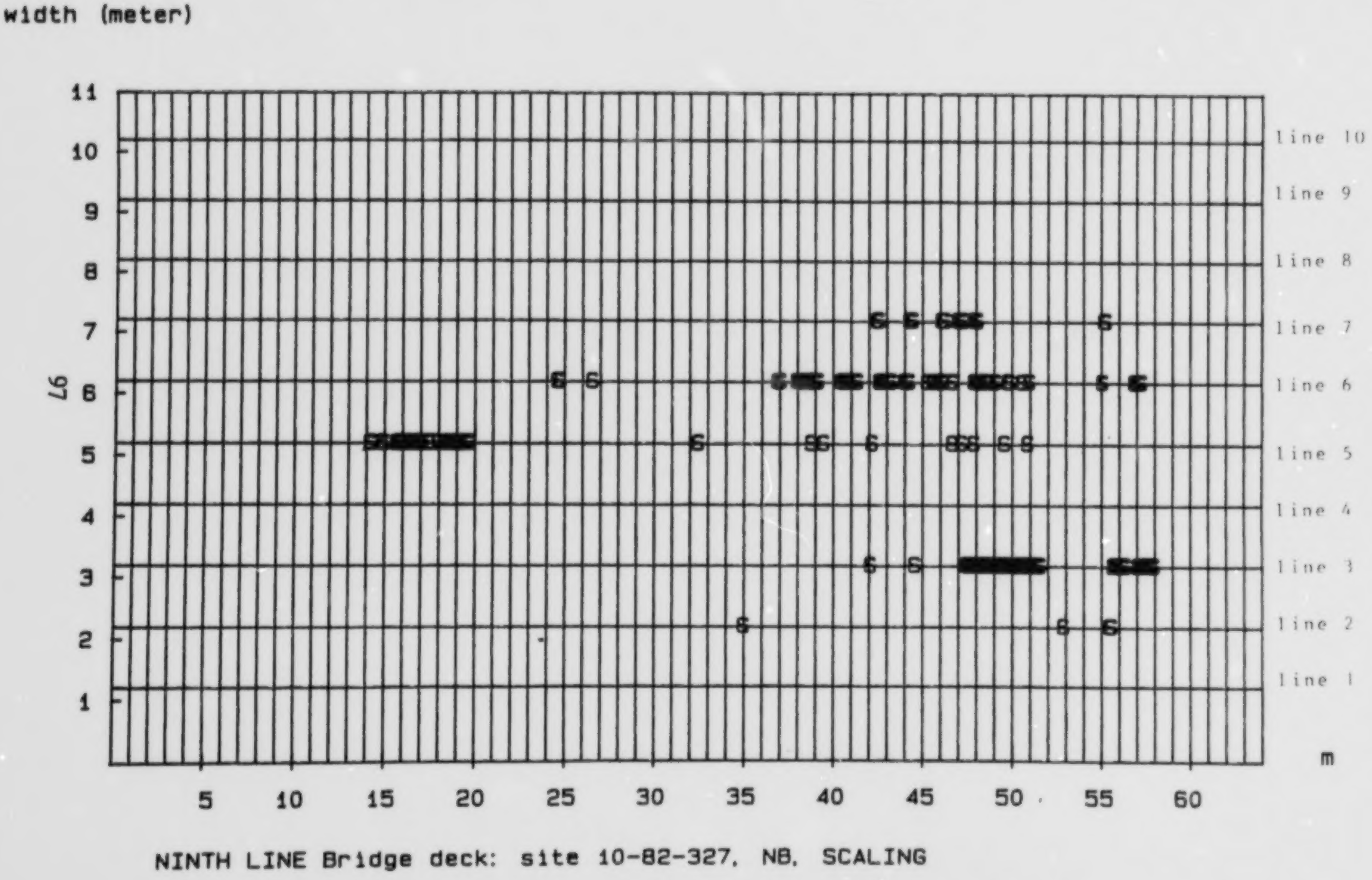
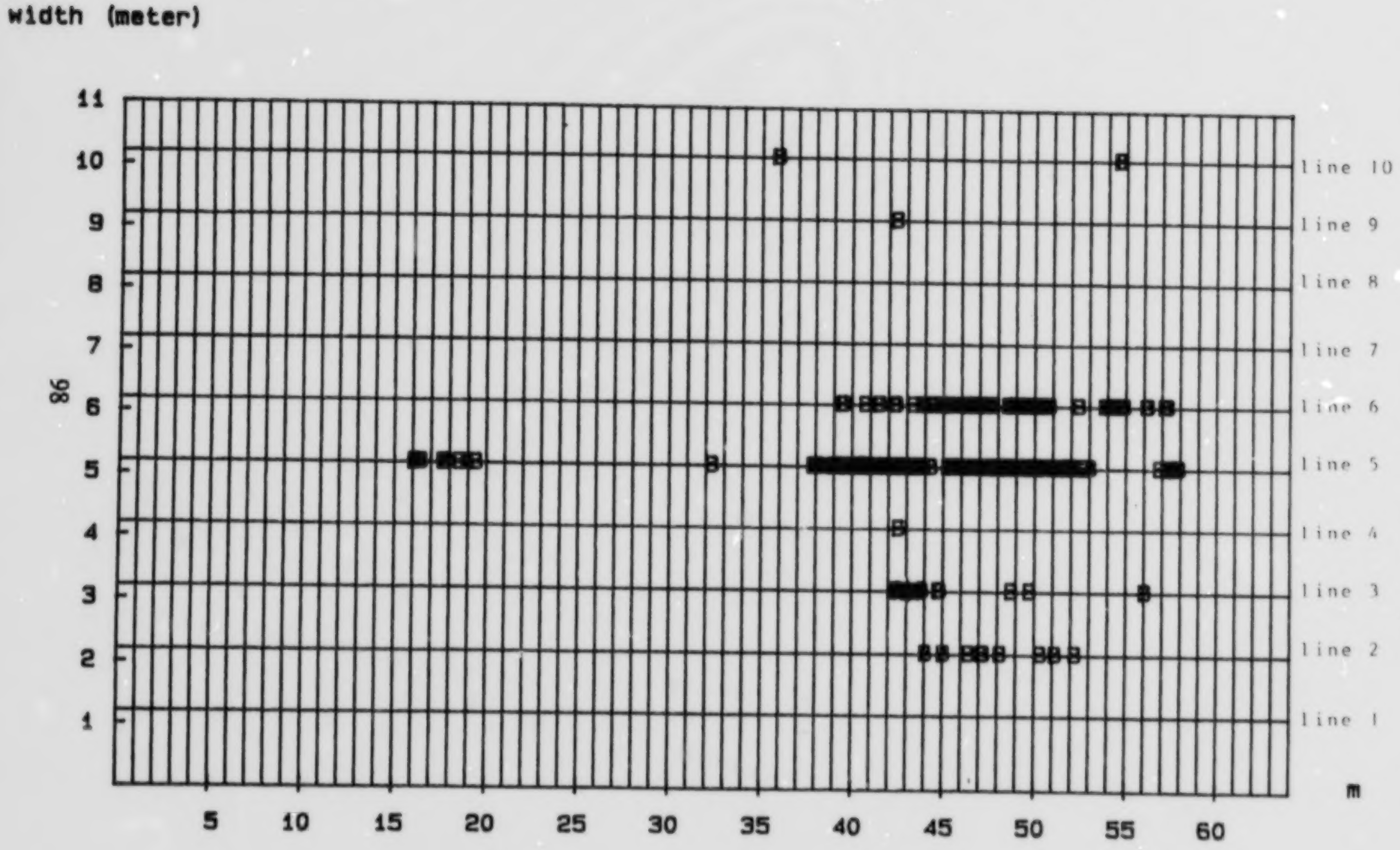


Figure 37: Scaled Areas at Ninth Line Detected by 386-DART.



NINTH LINE Bridge deck: site 10-82-327. NB. DEBONDING

Figure 38: Debonded Areas at Ninth Line Detected by 386-DART.

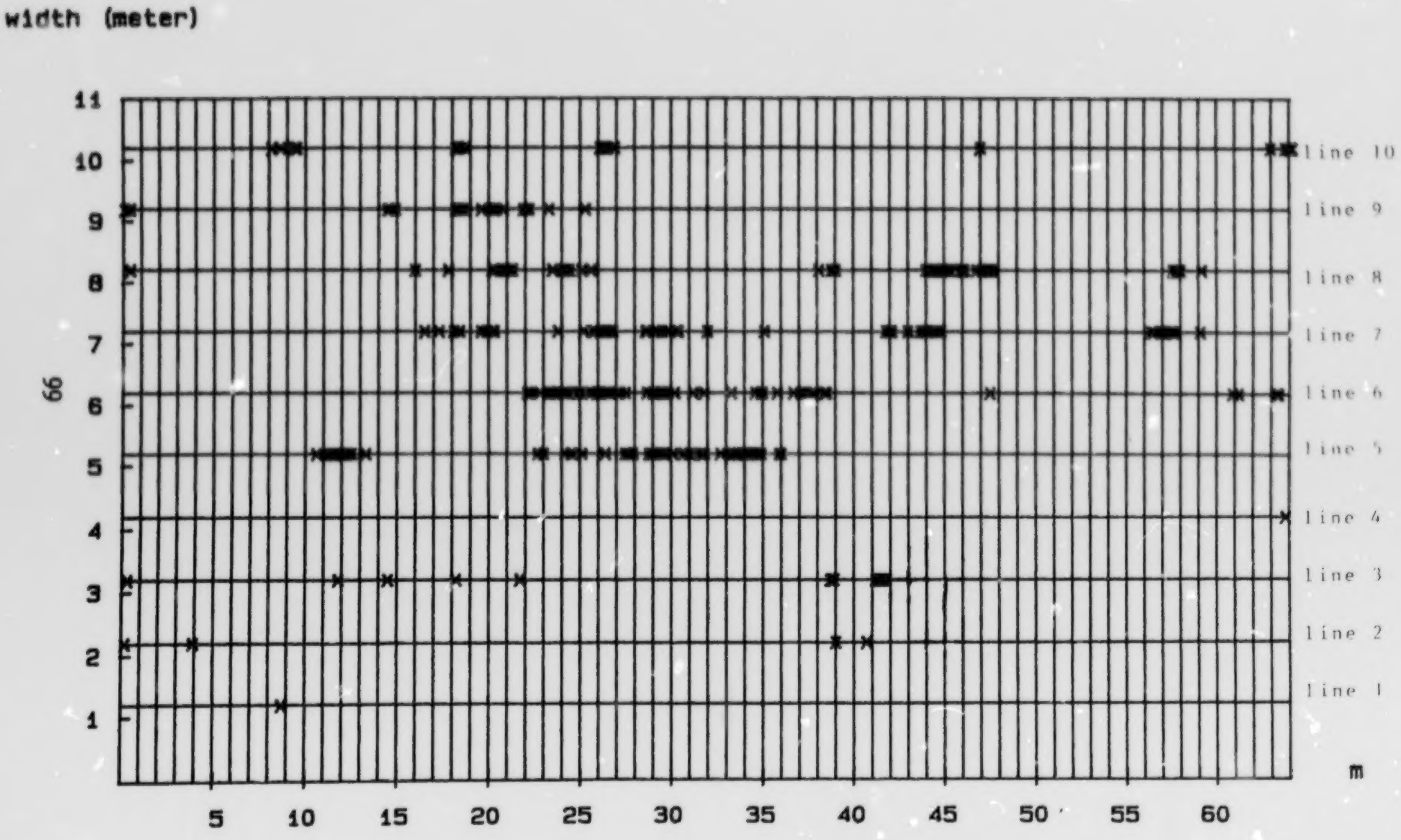
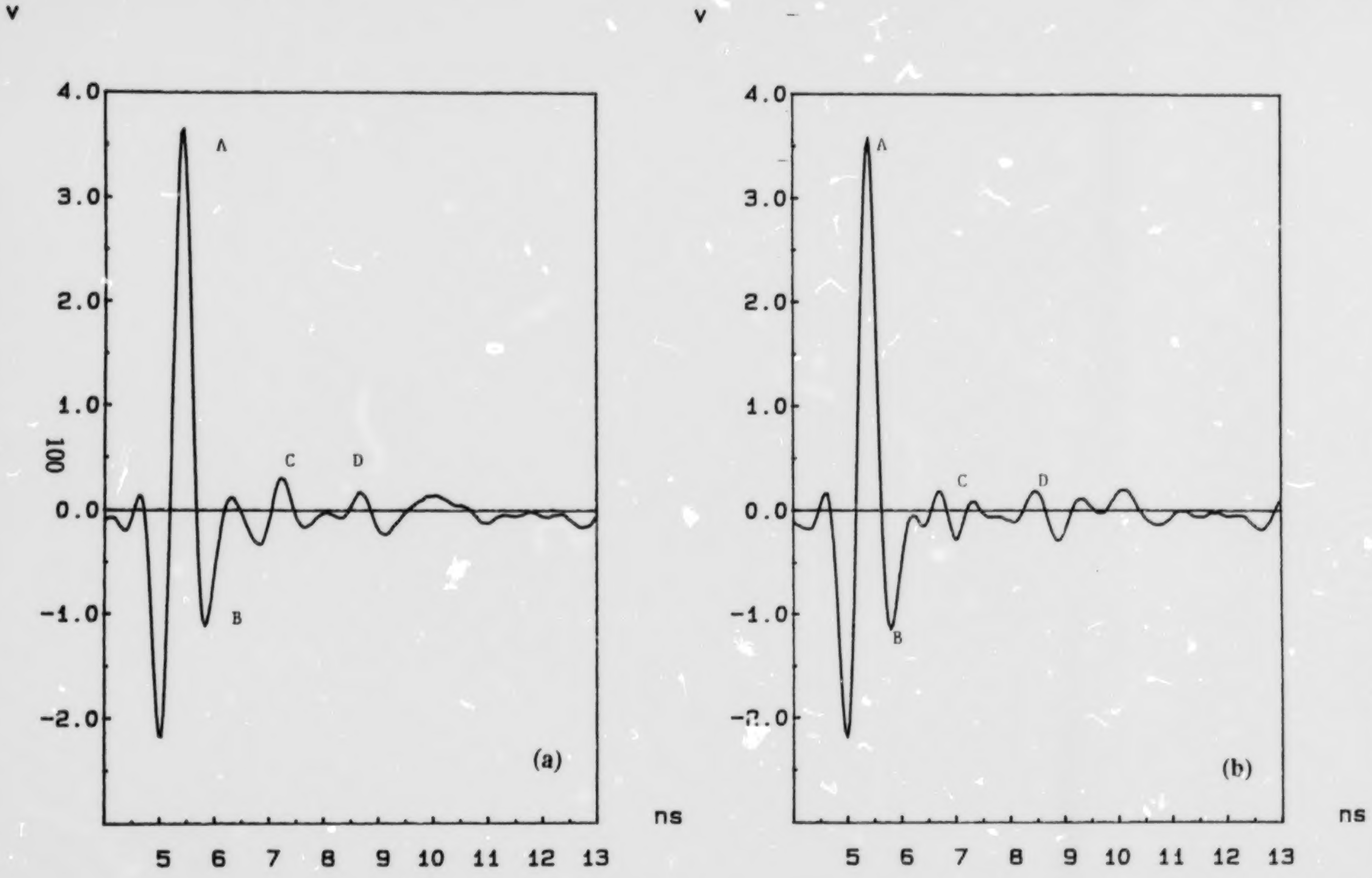


Figure 39: Delaminated Areas at Ninth Line Detected by 386-DART.



NINTH LINE: line 4 at 12m

line 4 at 18m

Figure 40: Ninth Line Waveforms from Line 4: (a) at 12m Indicates Good Condition. (b) at 18m Indicates Good Condition.

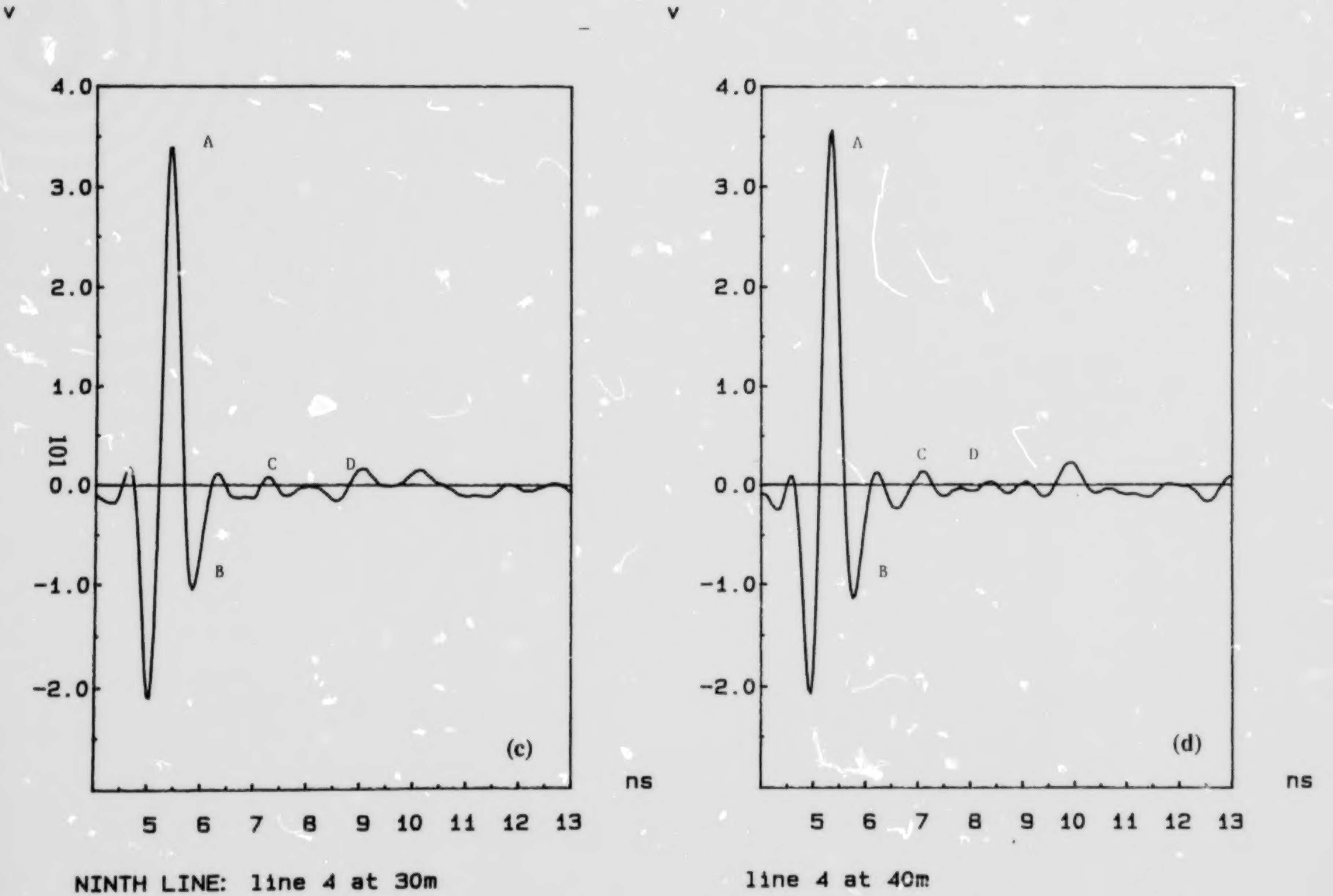
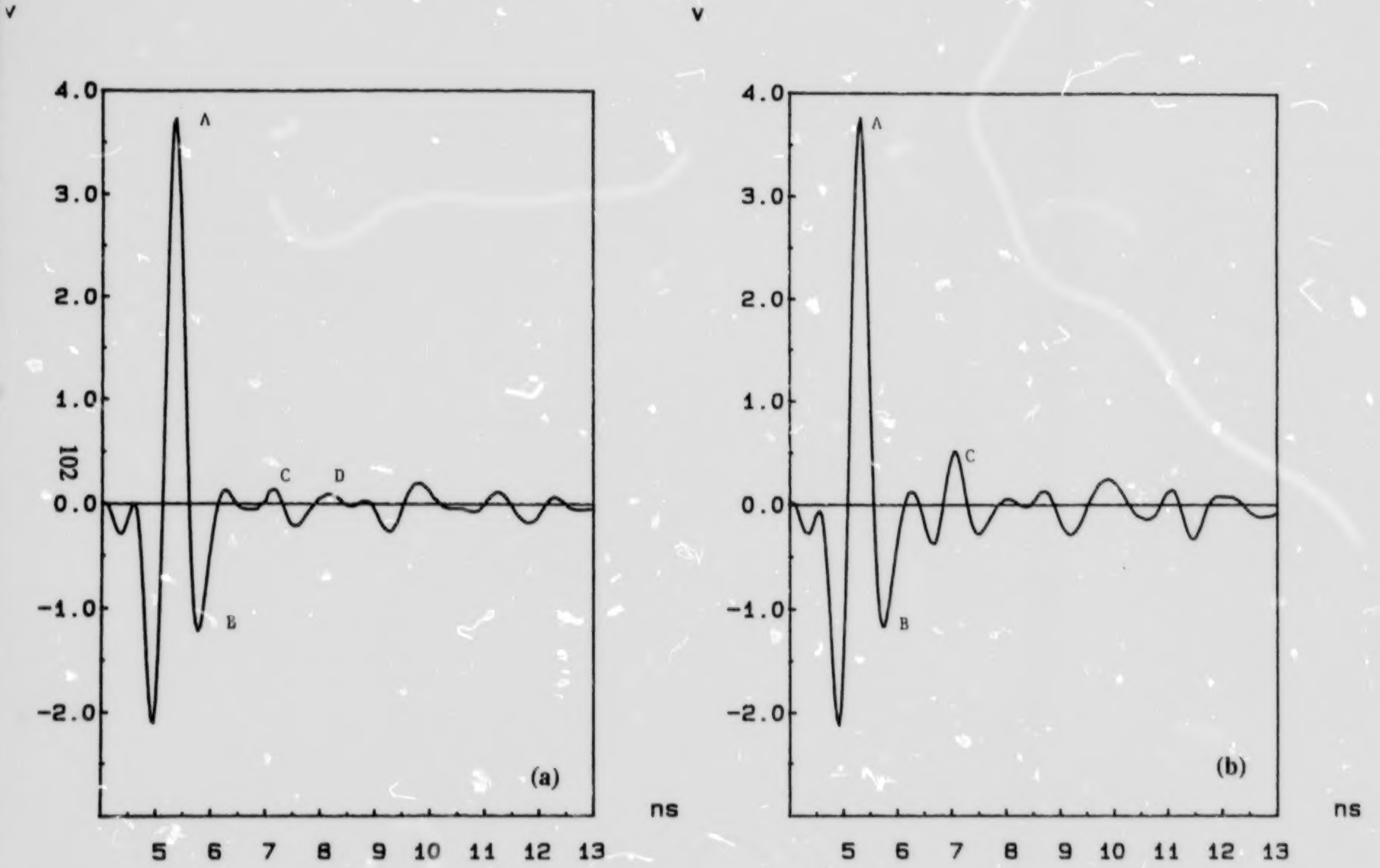


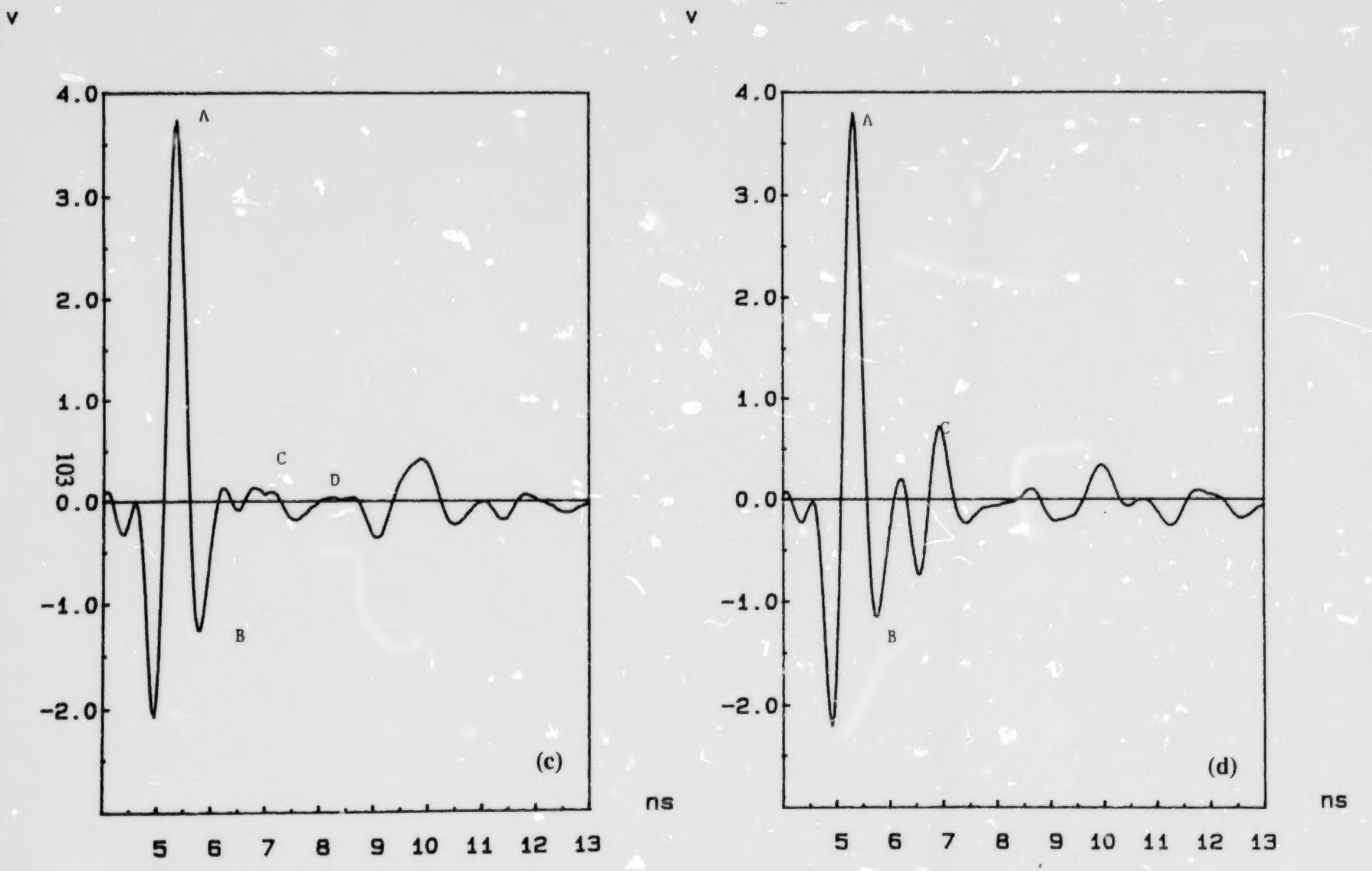
Figure 40: Ninth Line Waveforms from Line 4: (c) at 30m Indicates Good Condition. (d) at 40m Indicates Good Condition.



NINTH LINE: line 5 at 12m

line 5 at 18m

Figure 41: Ninth Line Waveforms from Line 5: (a) at 12m Indicates Delamination. (b) at 18m Indicates Debonding.



NINTH LINE: line 5 at 30m

line 5 at 40m

Figure 41: Ninth Line Waveforms from Line 5 (c) at 30m Indicates Delamination. (d) at 40m Indicates Scaling.

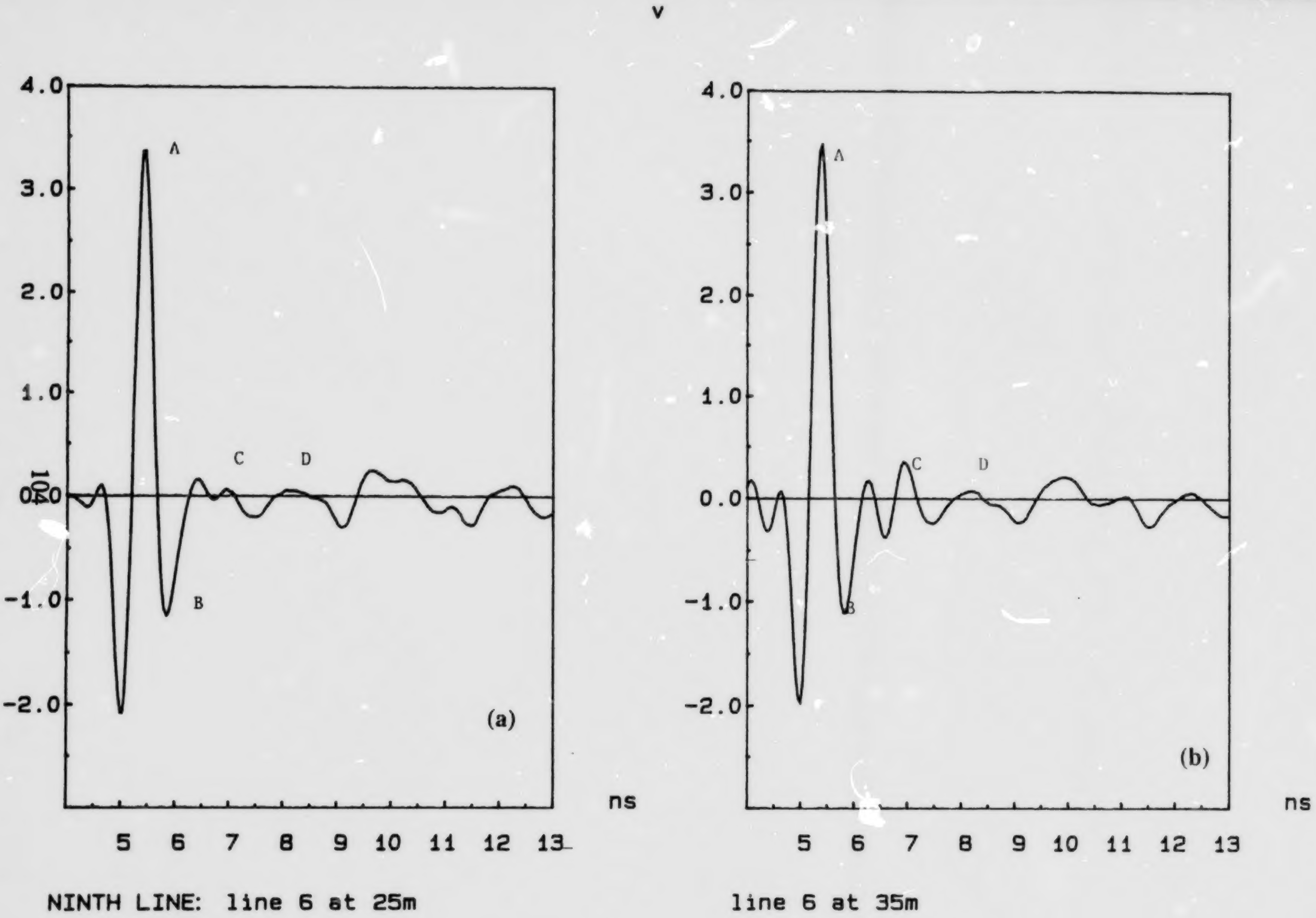
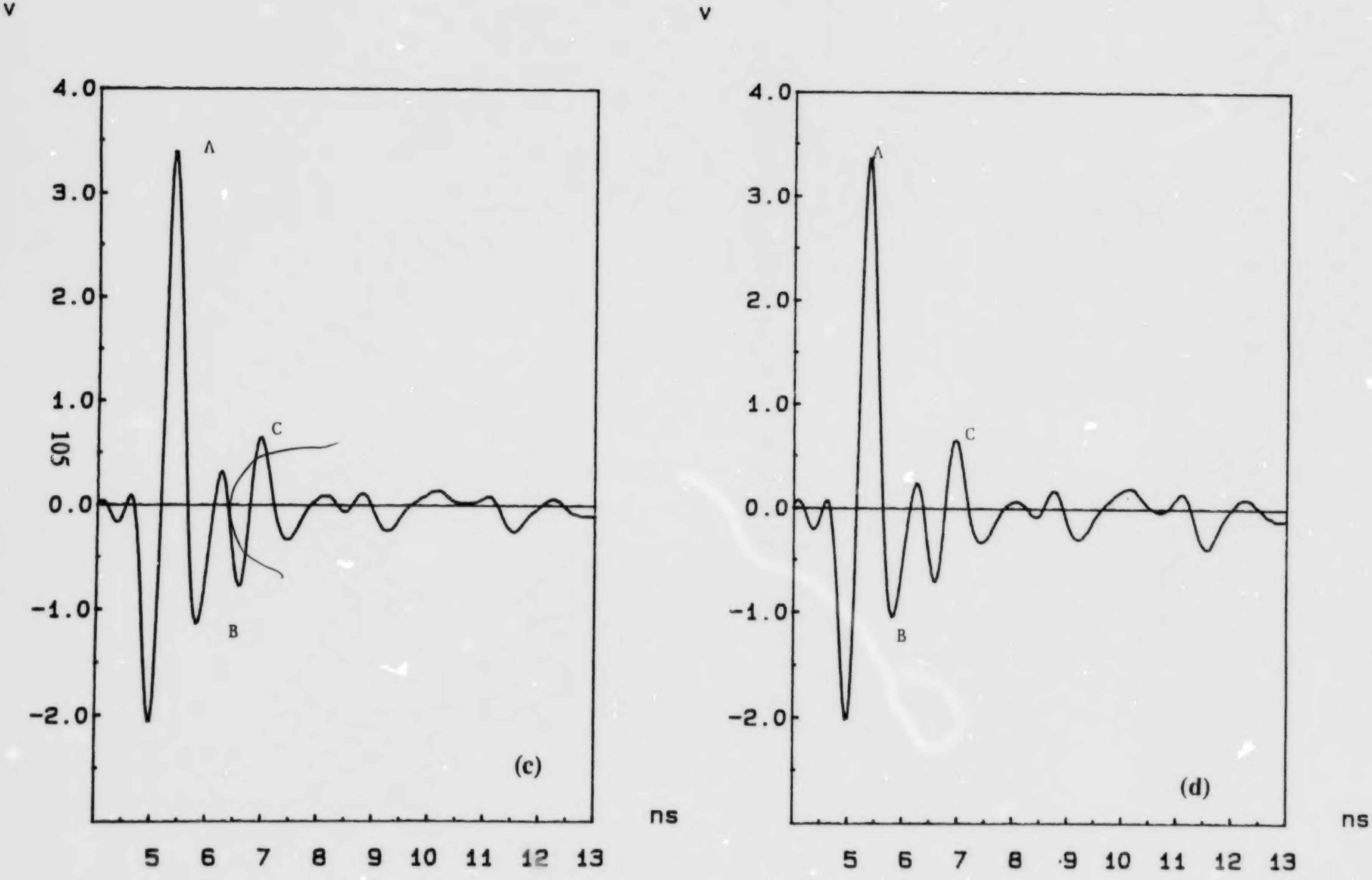


Figure 42: Ninth Line Waveforms from Line 6: (a) at 25m Indicates Delamination. (b) at 35m Indicates Delamination.



NINTH LINE: line 6 at 39m

line 6 at 45m

Figure 42: Ninth Line Waveforms from Line 6: (c) at 39m Indicates Scaling. (d) at 45m Indicates Debonding.

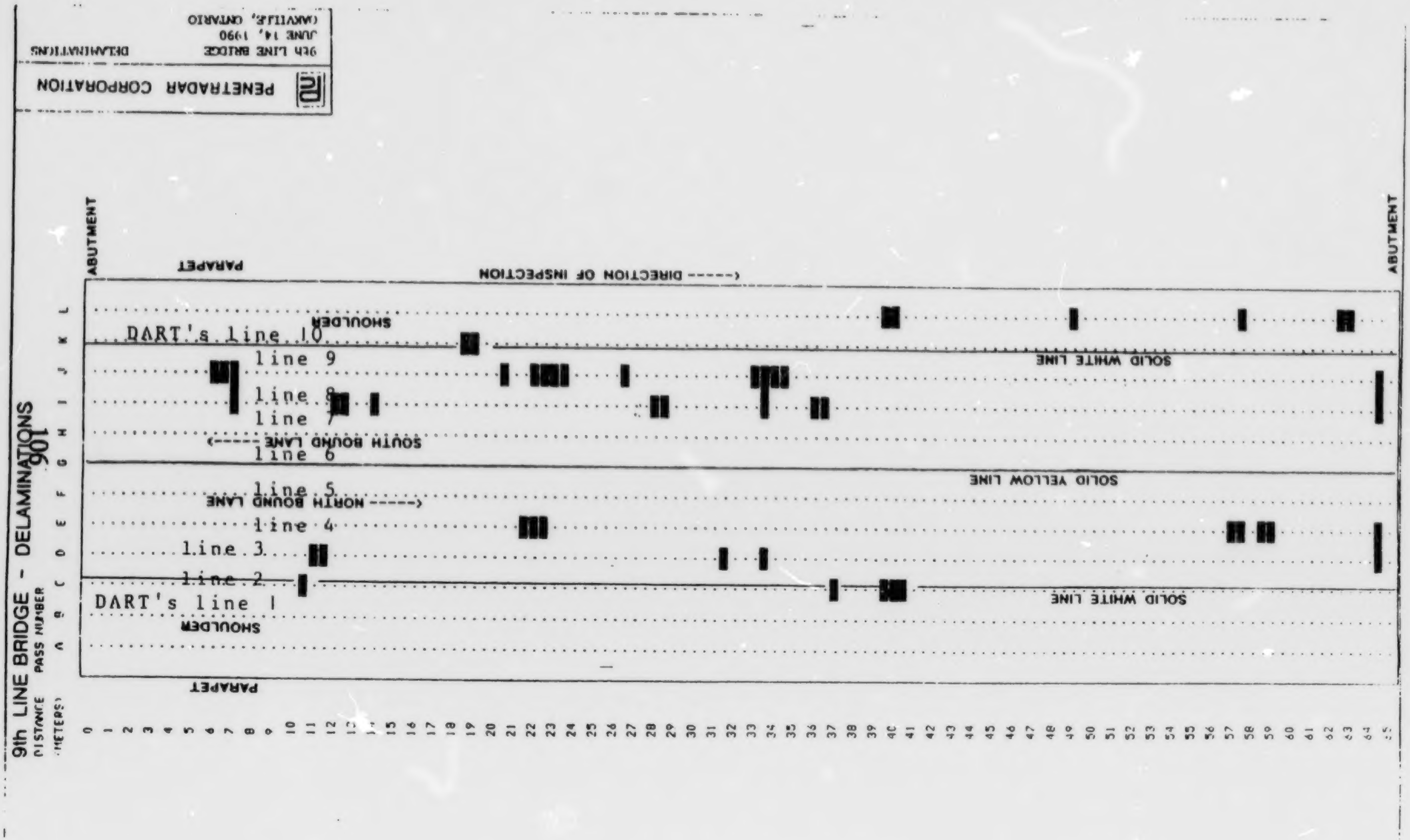


Figure 43: Delaminated Areas at Ninth Line Detected by Penetradar.

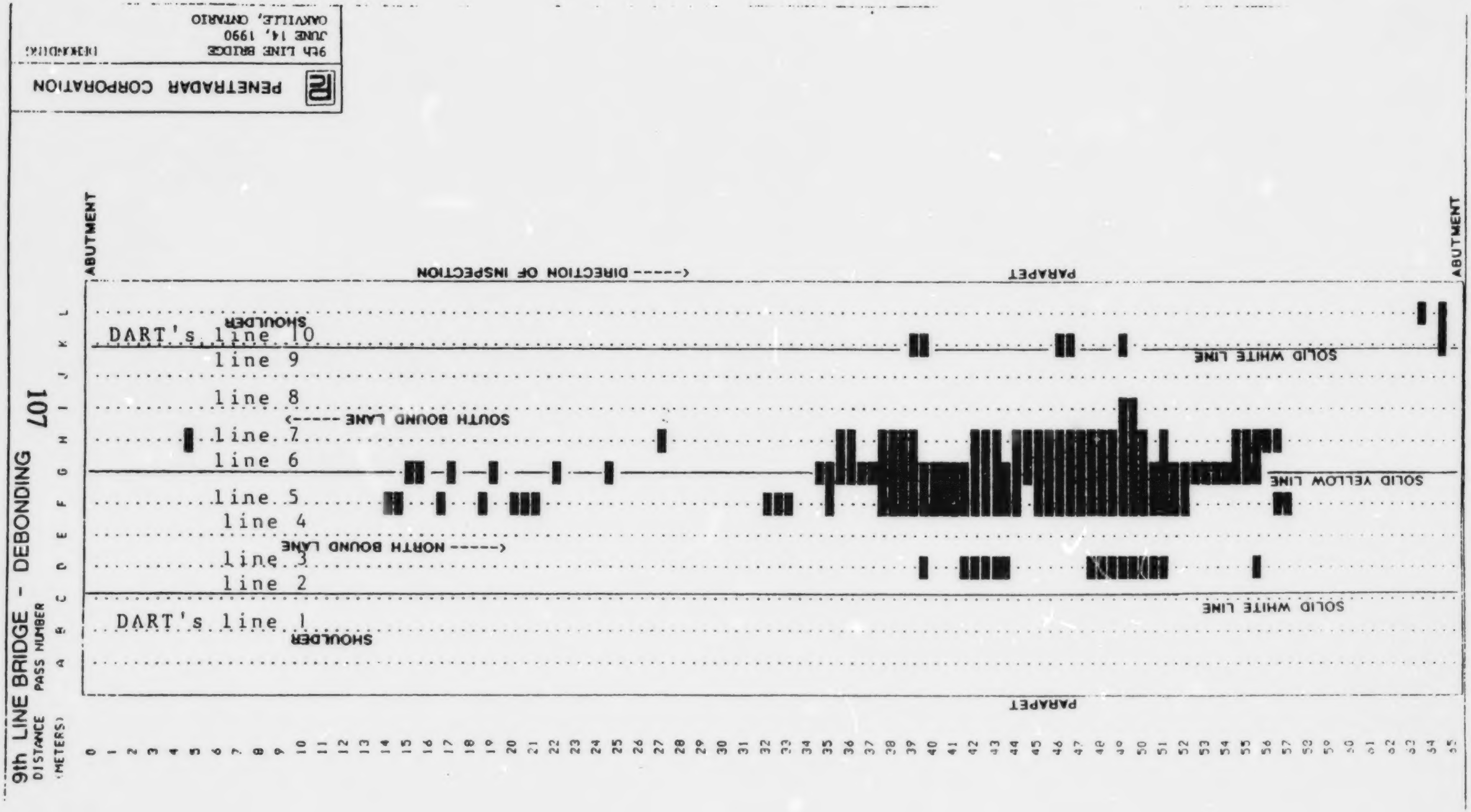
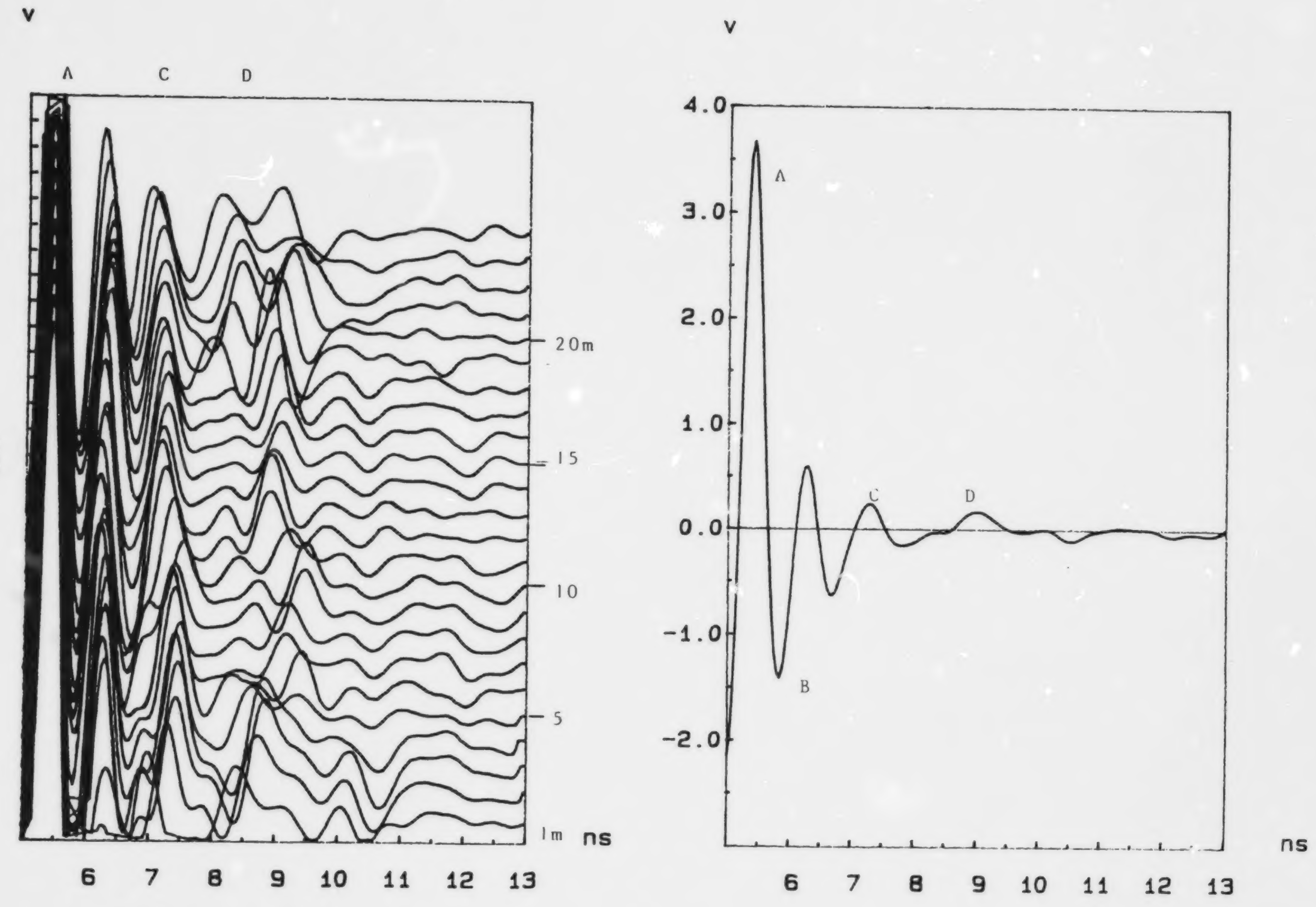


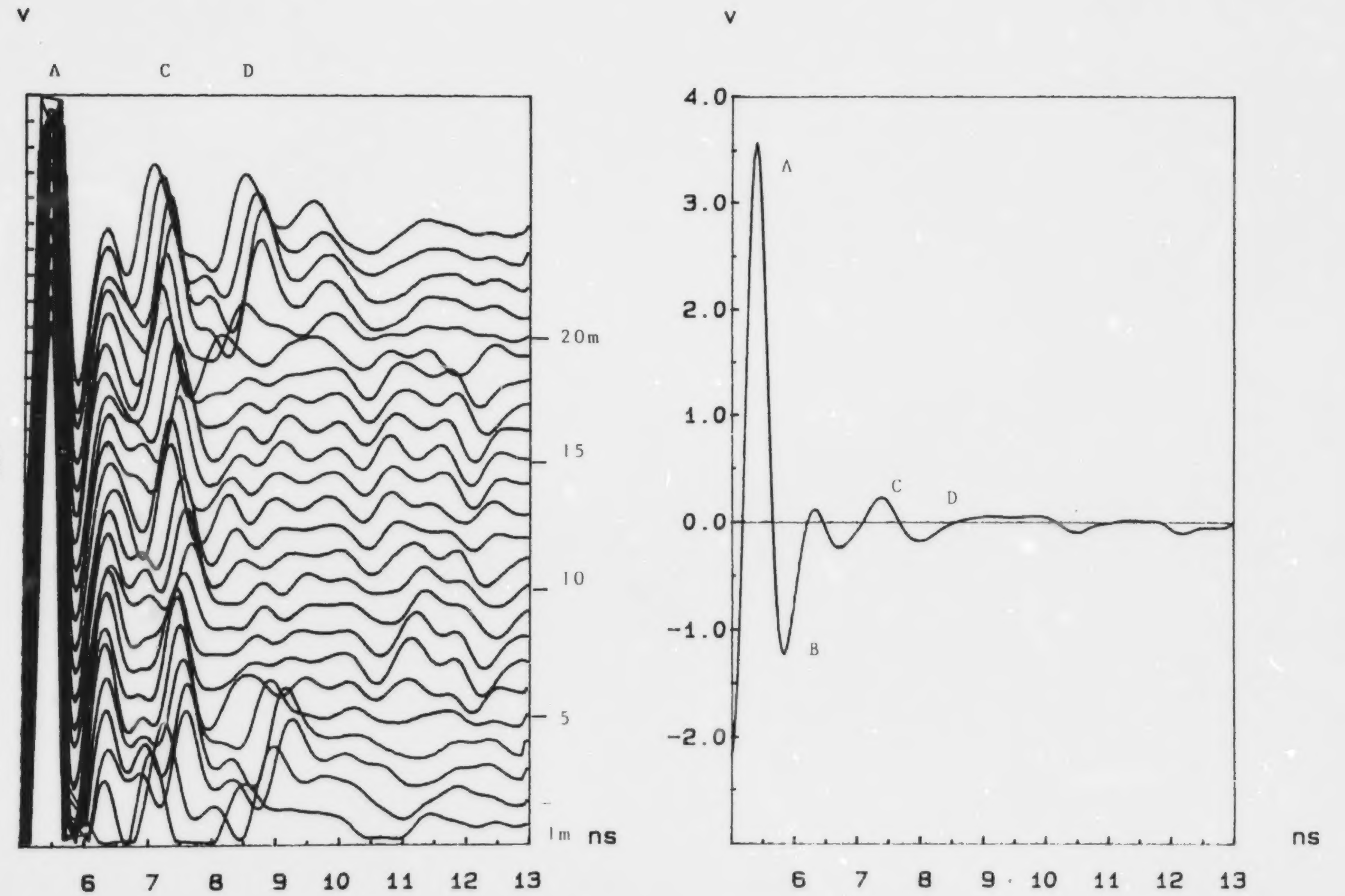
Figure 44: Debonded Areas at Ninth Line Detected by Penetradar.



Little Bob. site 32-079. line 1

average waveform

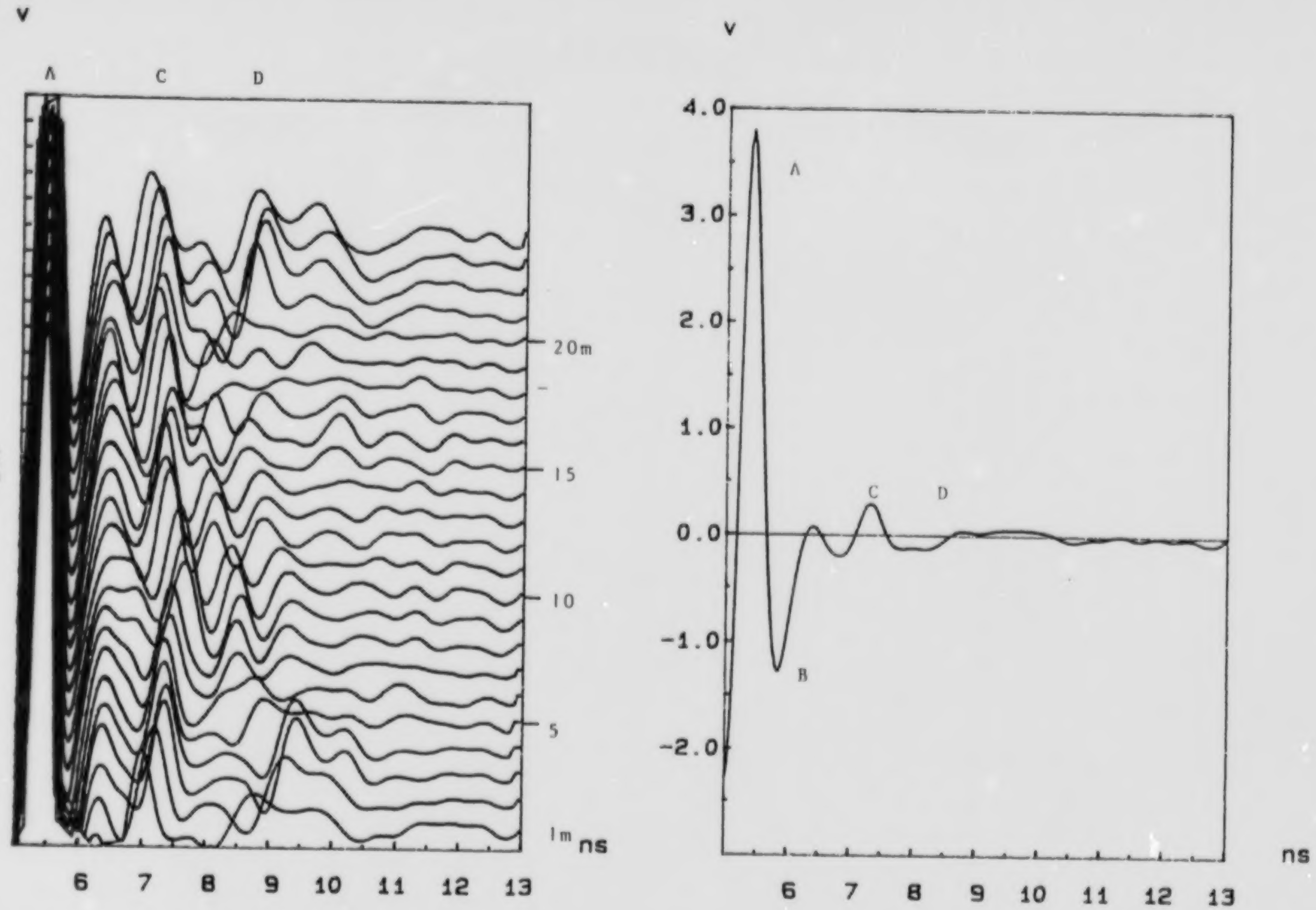
Figure 45: Waveforms from Line 1 of Little Bob South Bridge.



Little Bob. site 32-079. line 2

average waveform

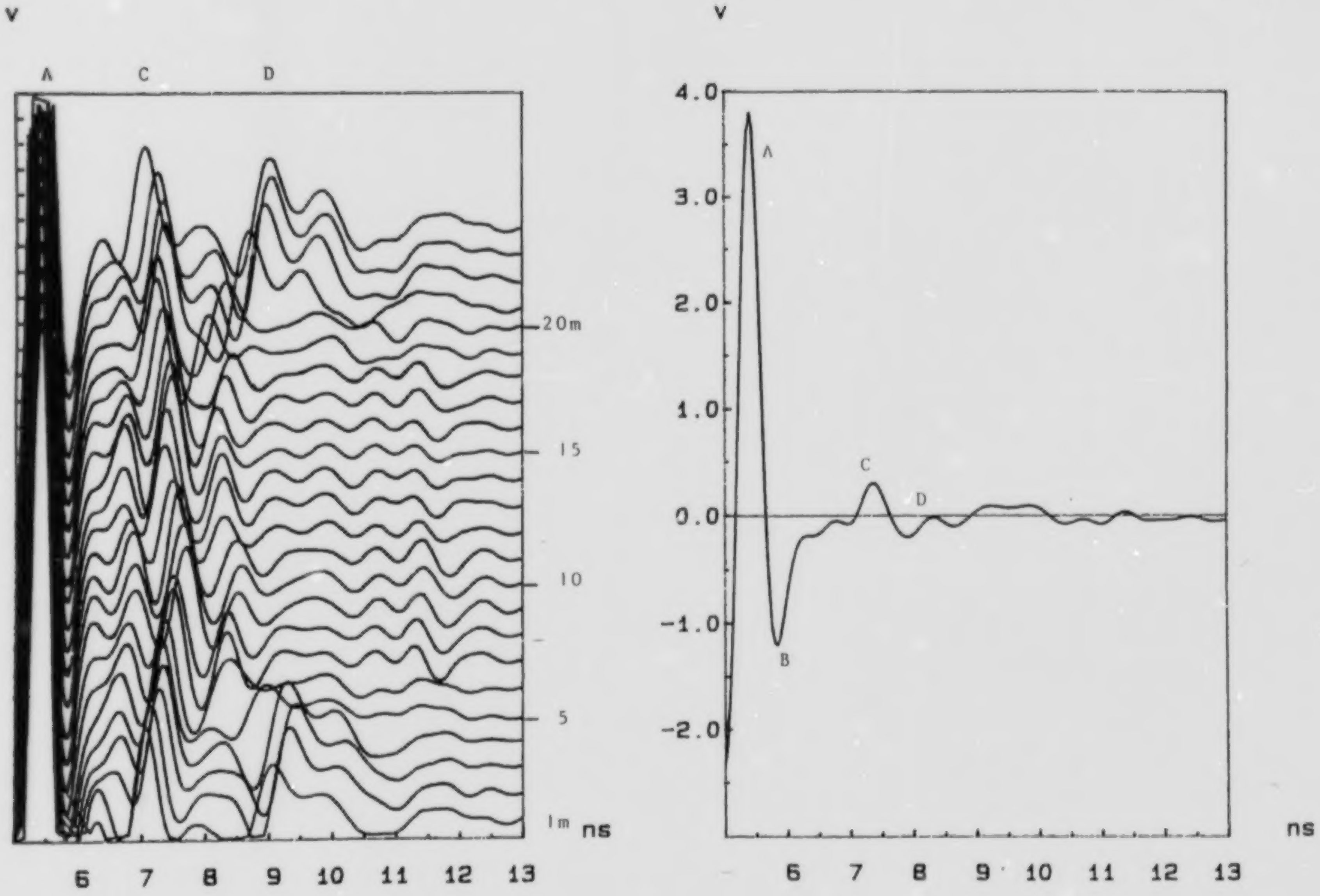
Figure 46: Waveforms from Line 2 of Little Bob South Bridge.



Little Bob. site 32-079. line 3

average waveform

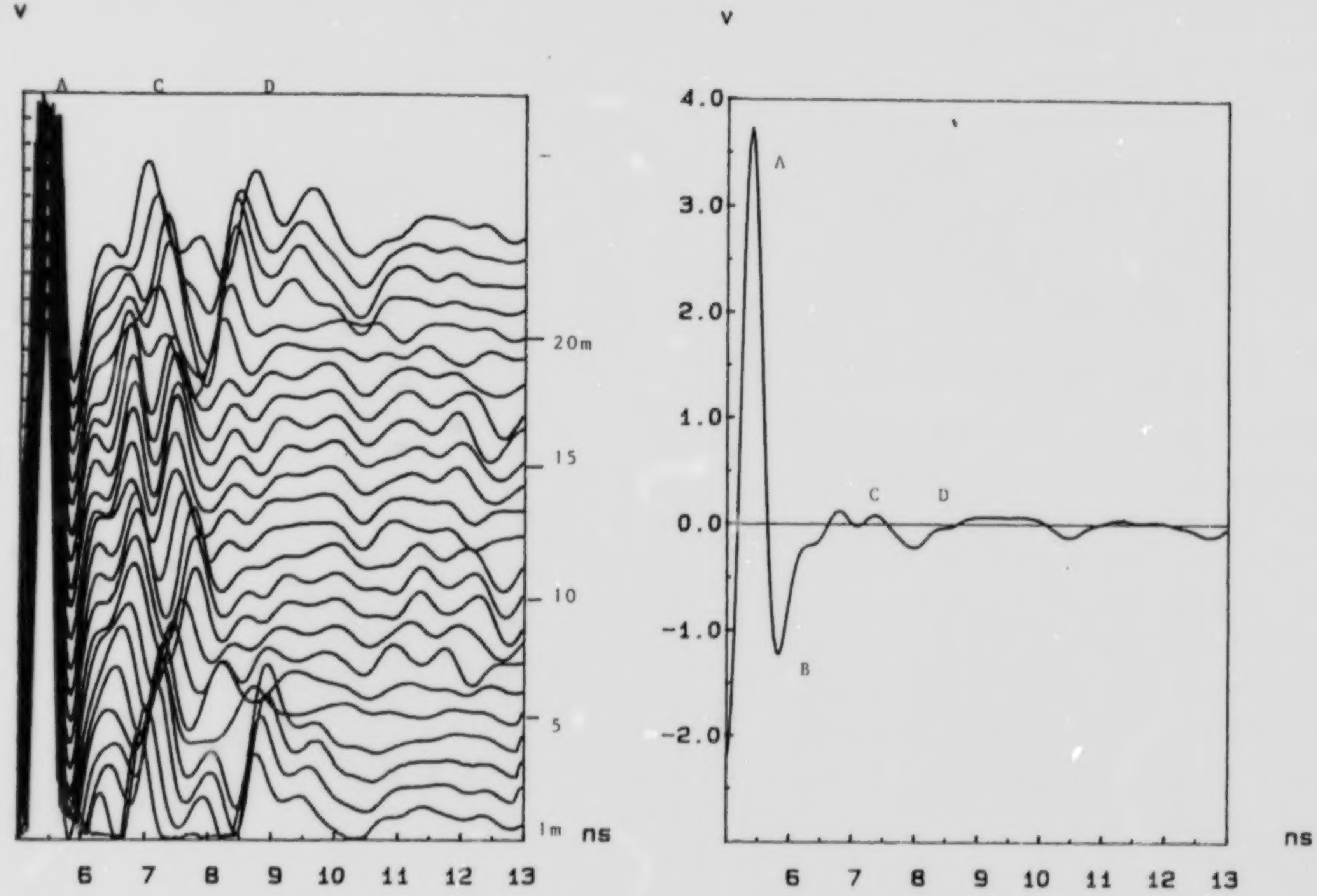
Figure 47: Waveforms from Line 3 of Little Bob South Bridge.



Little Bob. site 32-079. line 4

average waveform

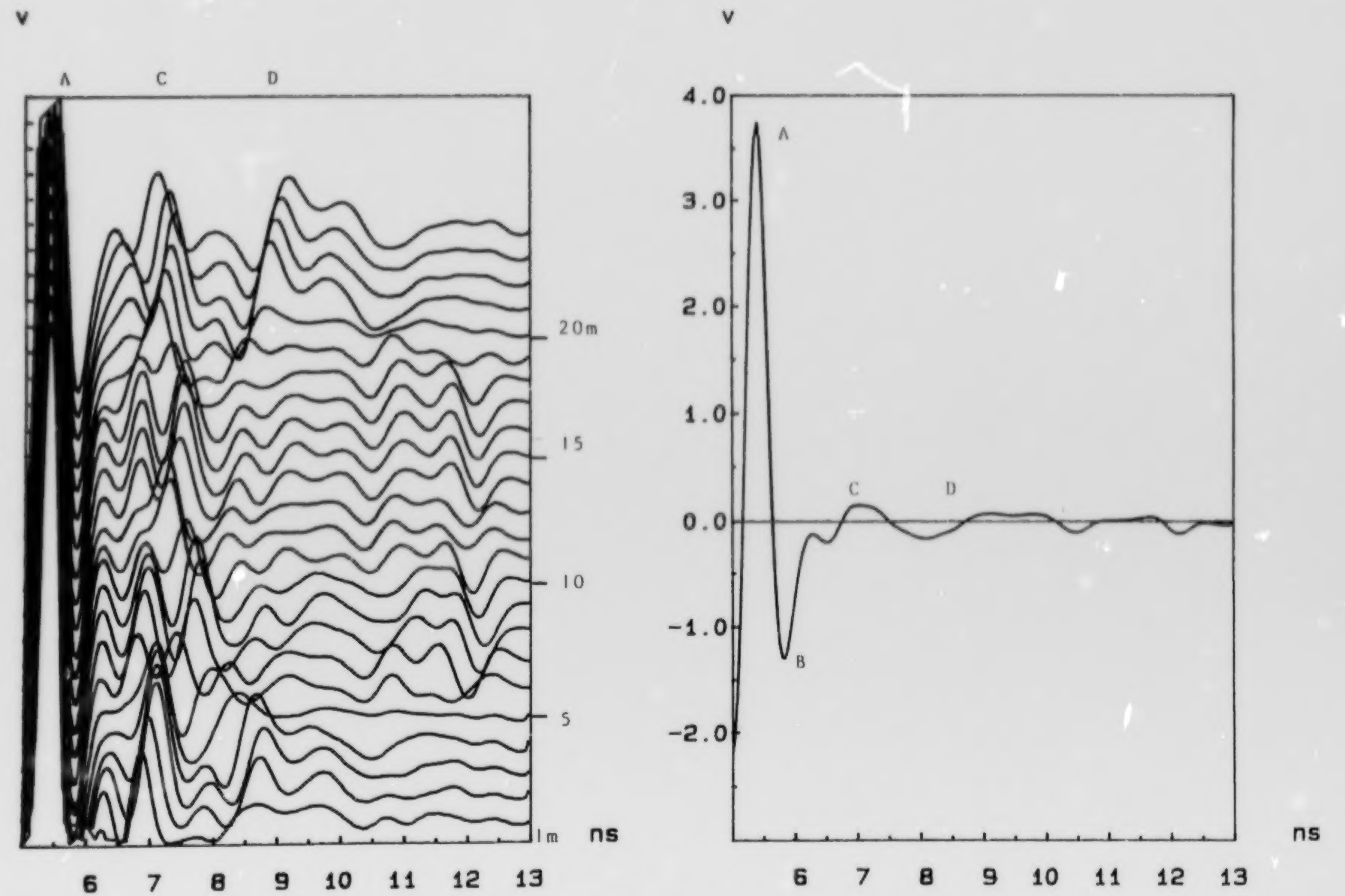
Figure 48: Waveforms from Line 4 of Little Bob South Bridge.



Little Bob, site 32-079, line 5

average waveform

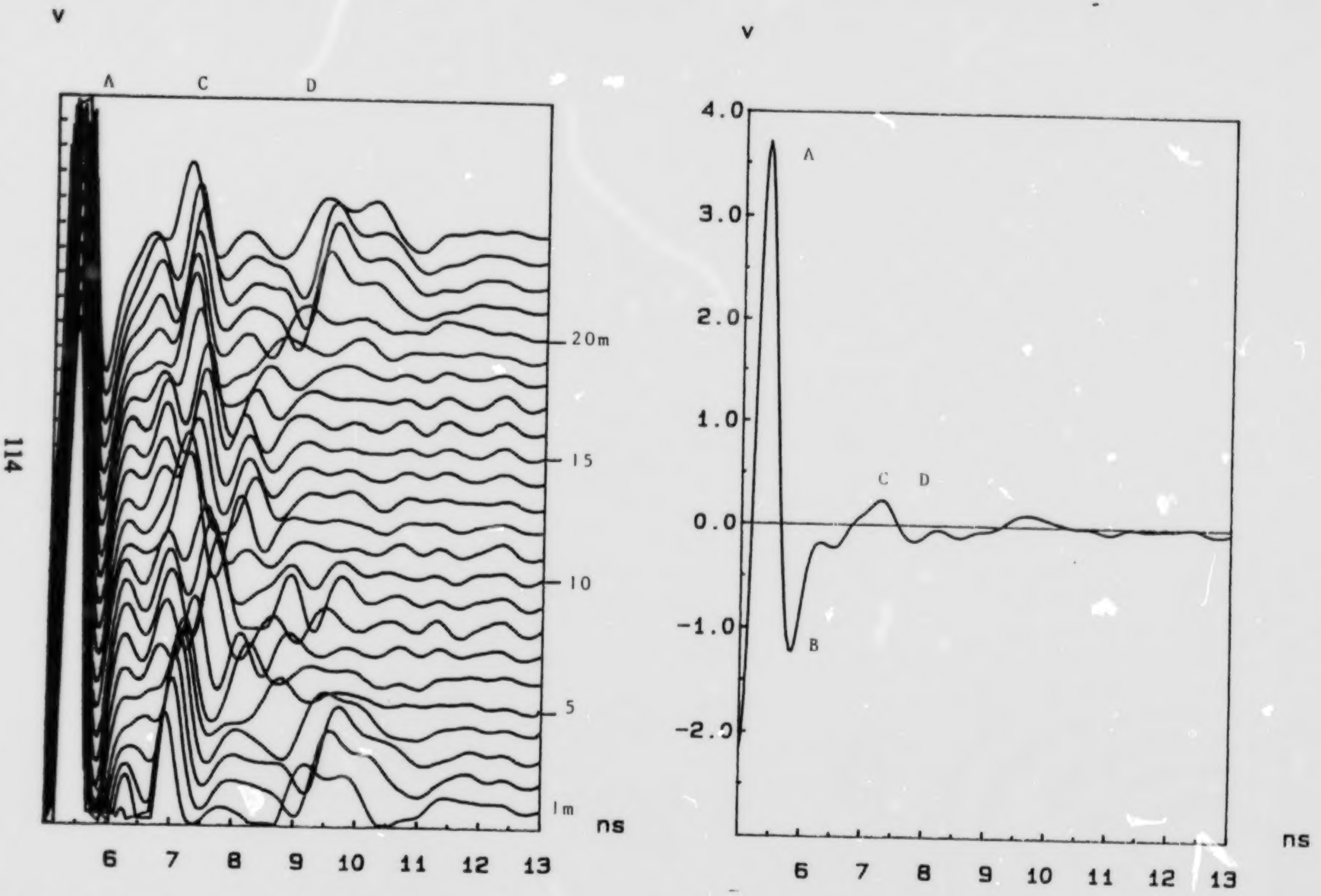
Figure 49: Waveforms from Line 5 of Little Bob South Bridge.



Little Bob. site 32-079. line 6

average waveform

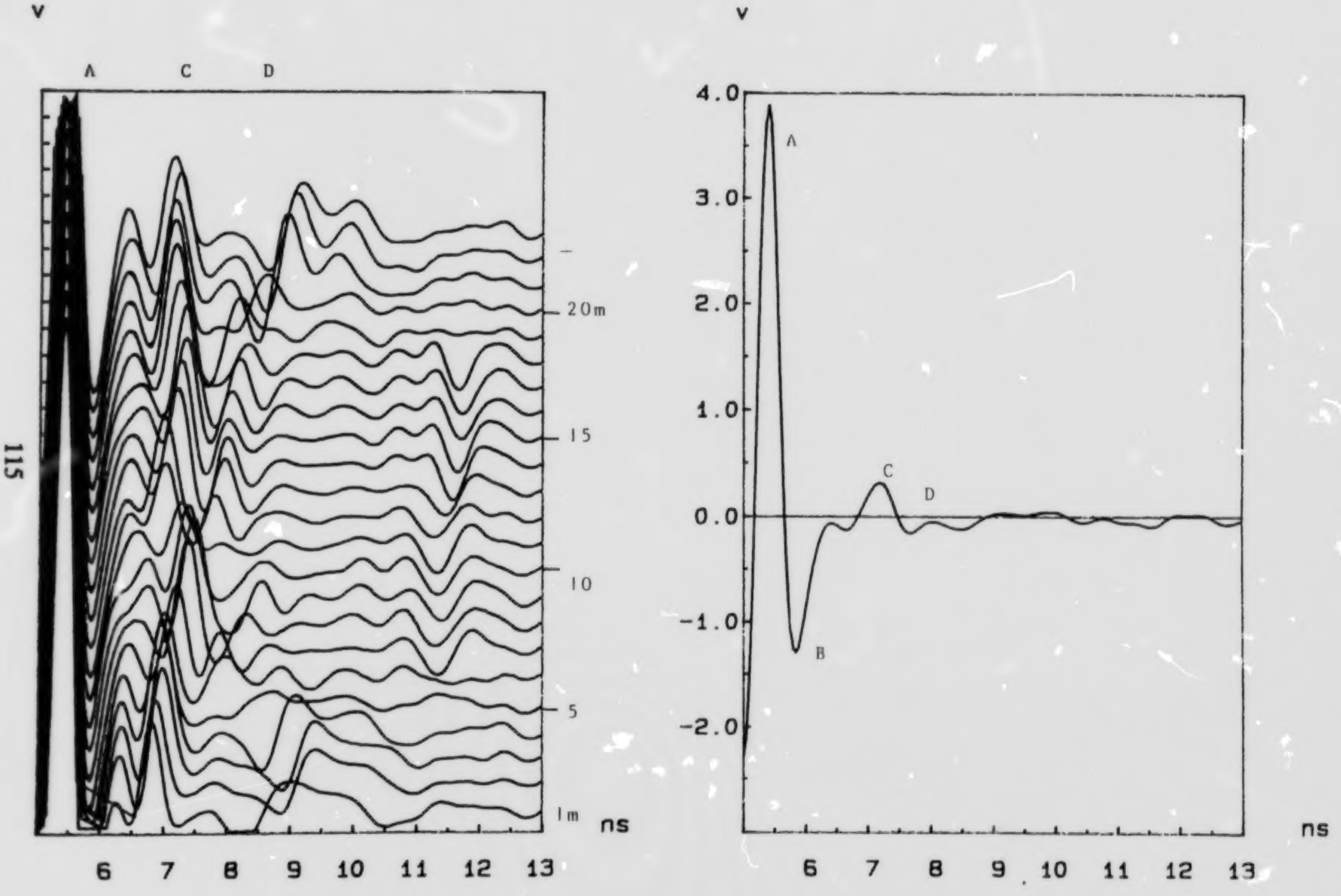
Figure 50: Waveforms from Line 6 of Little Bob South Bridge.



Little Bob. site 32-079. line 7

average waveform

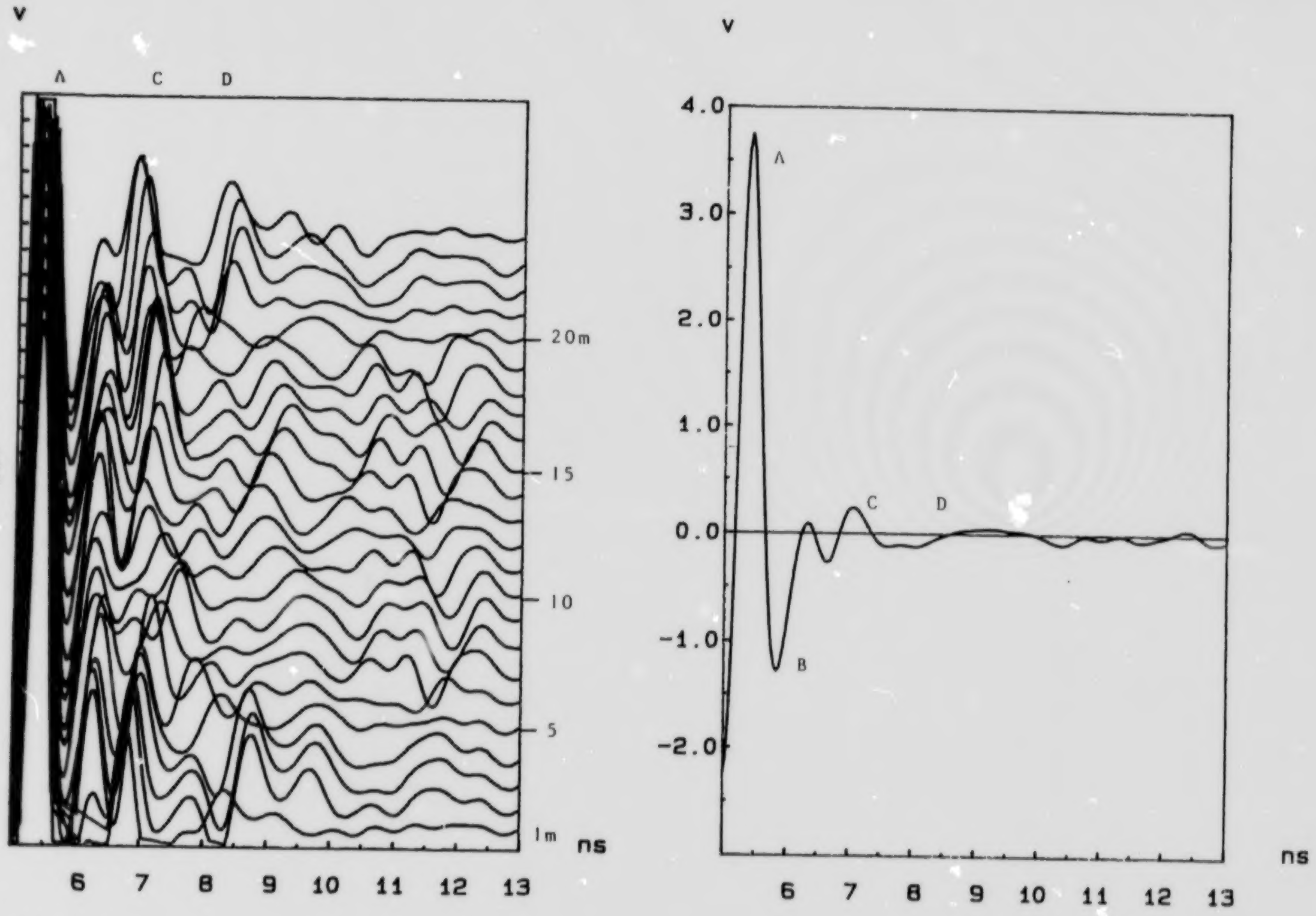
Figure 51: Waveforms from Line 7 of Little Bob South Bridge.



Little Bob, site 32-079, line 8

average waveform

Figure 52: Waveforms from Line 8 of Little Bob South Bridge.



Little Bob. site 32-079. line 9

average waveform

Figure 53: Waveforms from Line 9 of Little Bob South Bridge.

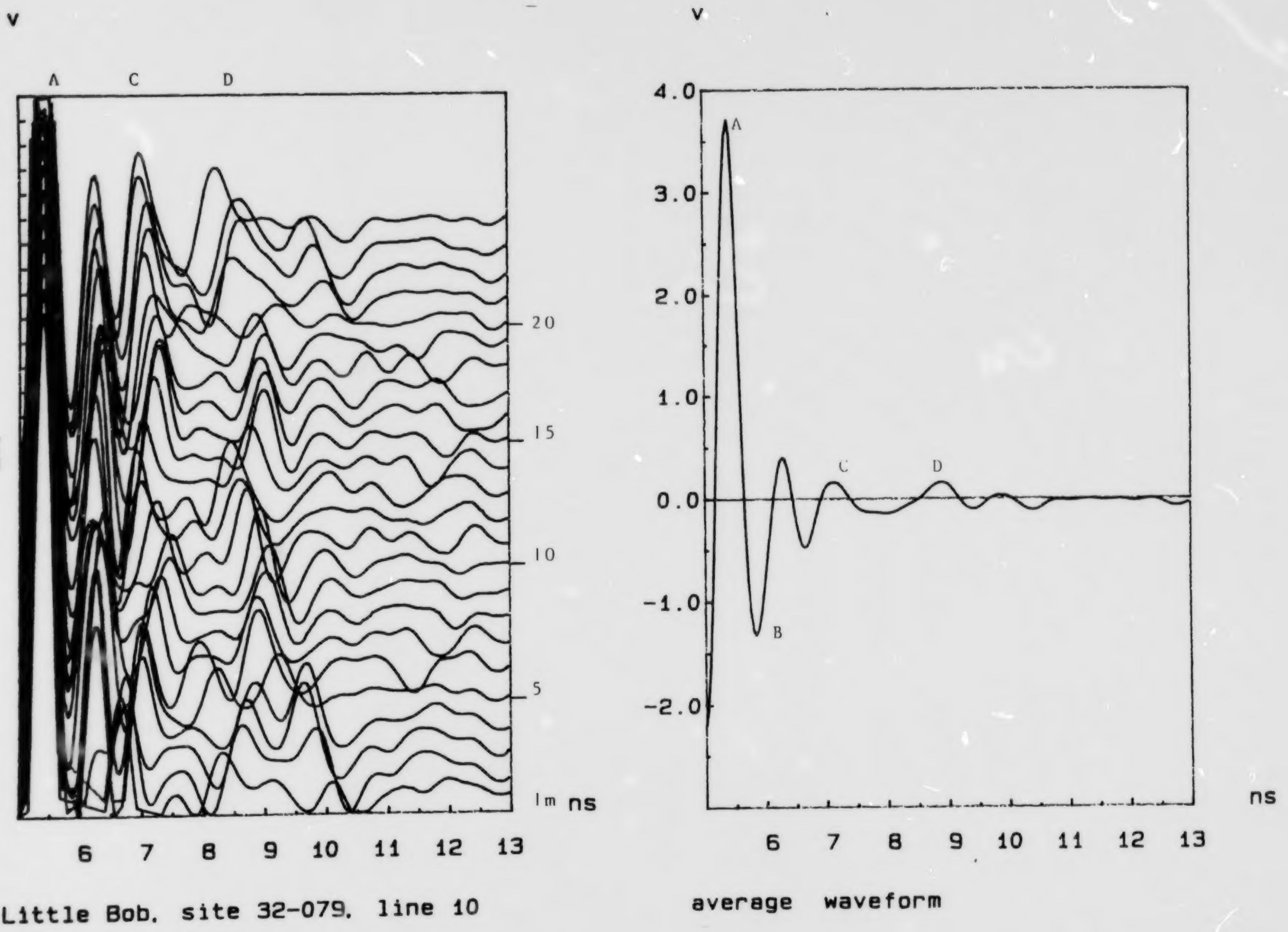
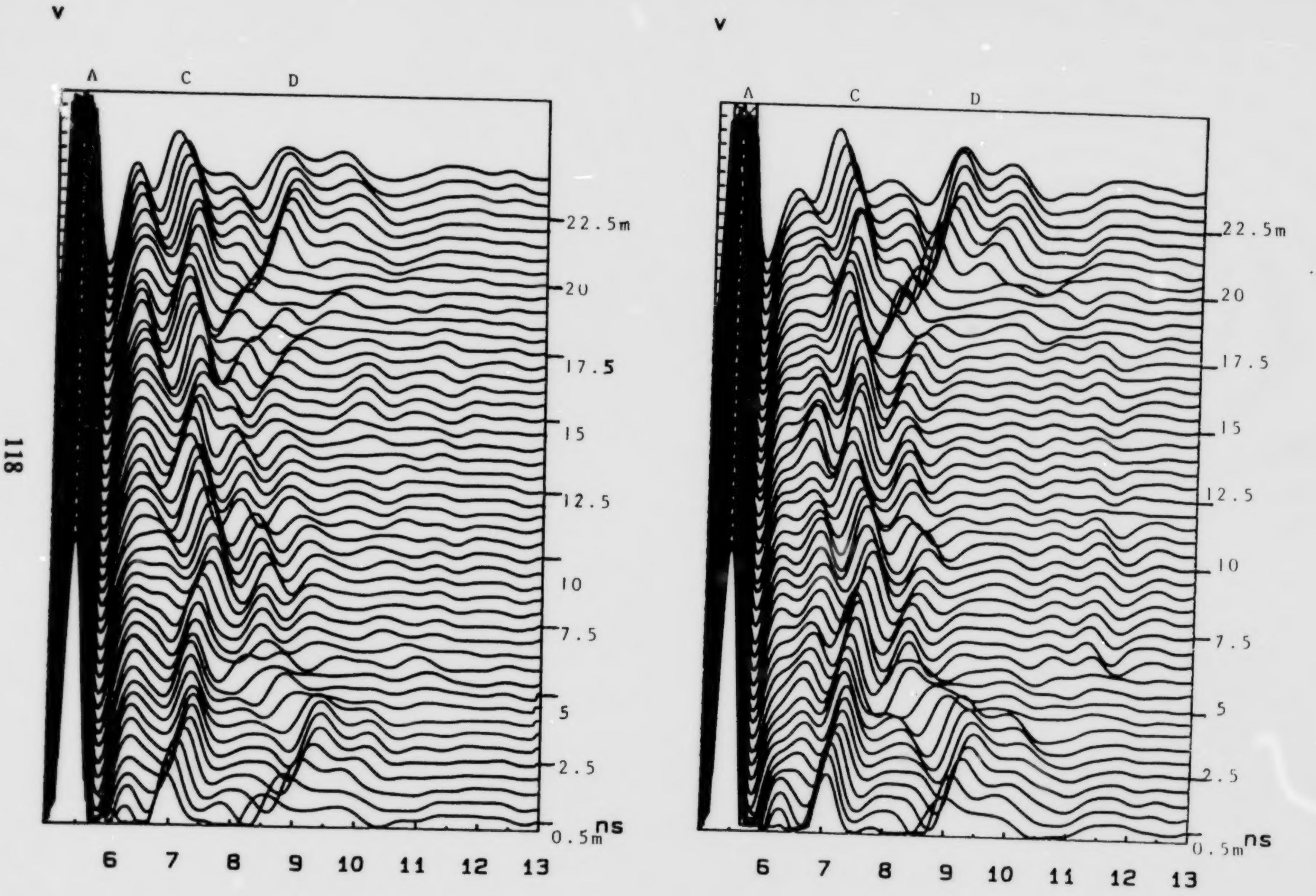


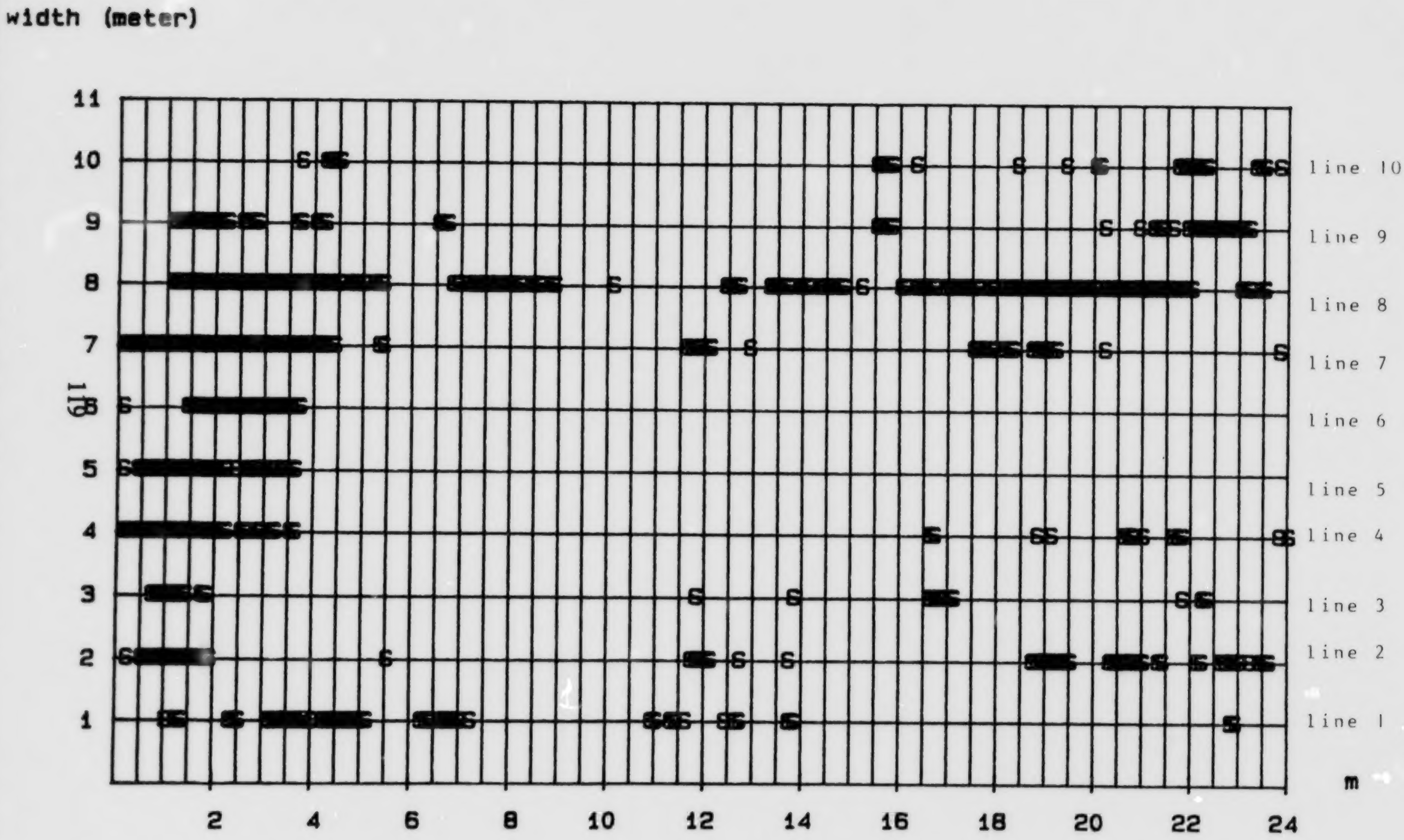
Figure 54: Waveforms from Line 10 of Little Bob South Bridge.



Little Bob. site 32-079. line 3

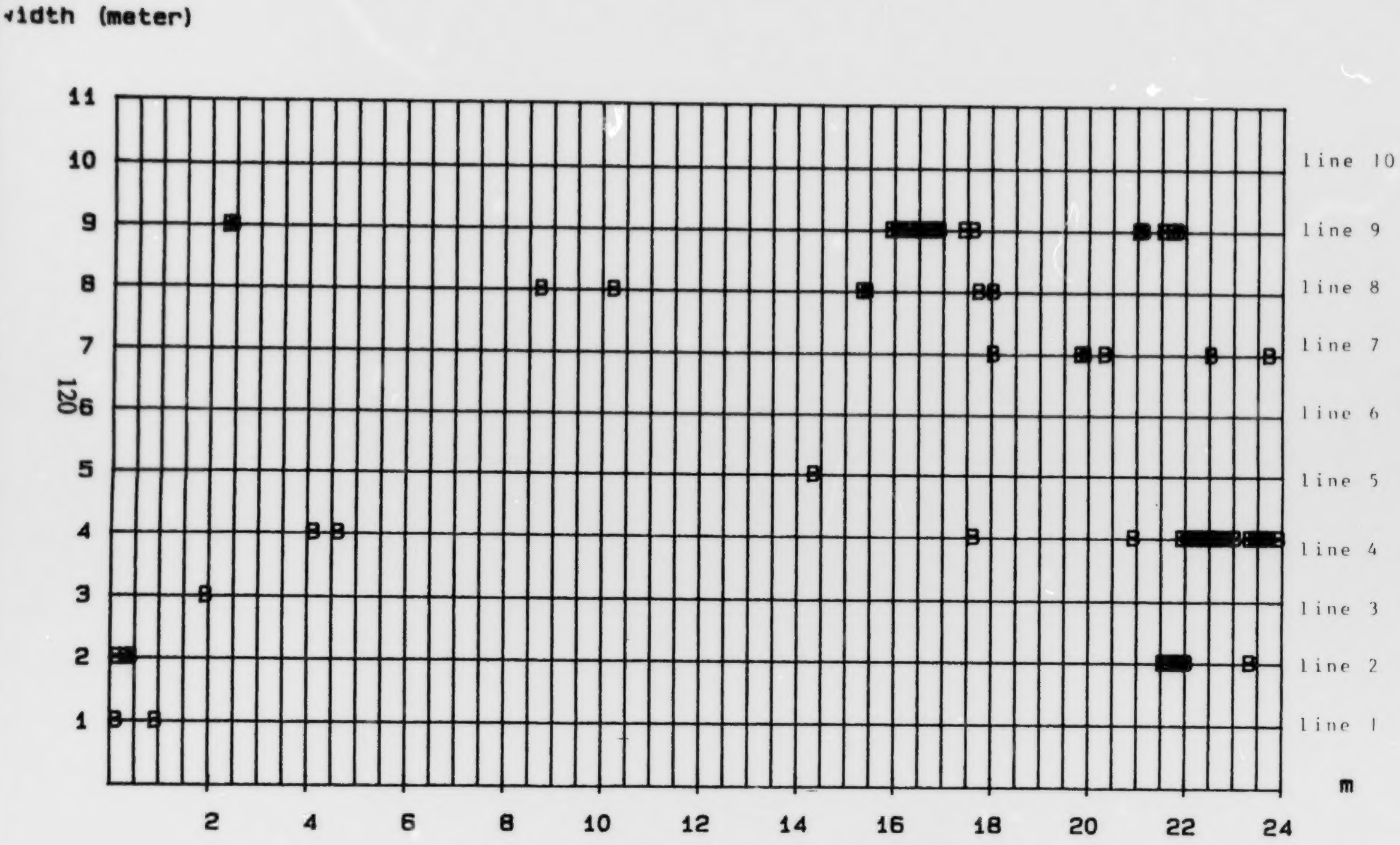
Little Bob. site 32-079. line 4

Figure 55: Detailed Profile for Lines 3 and 4 of Little Bob South Bridge.



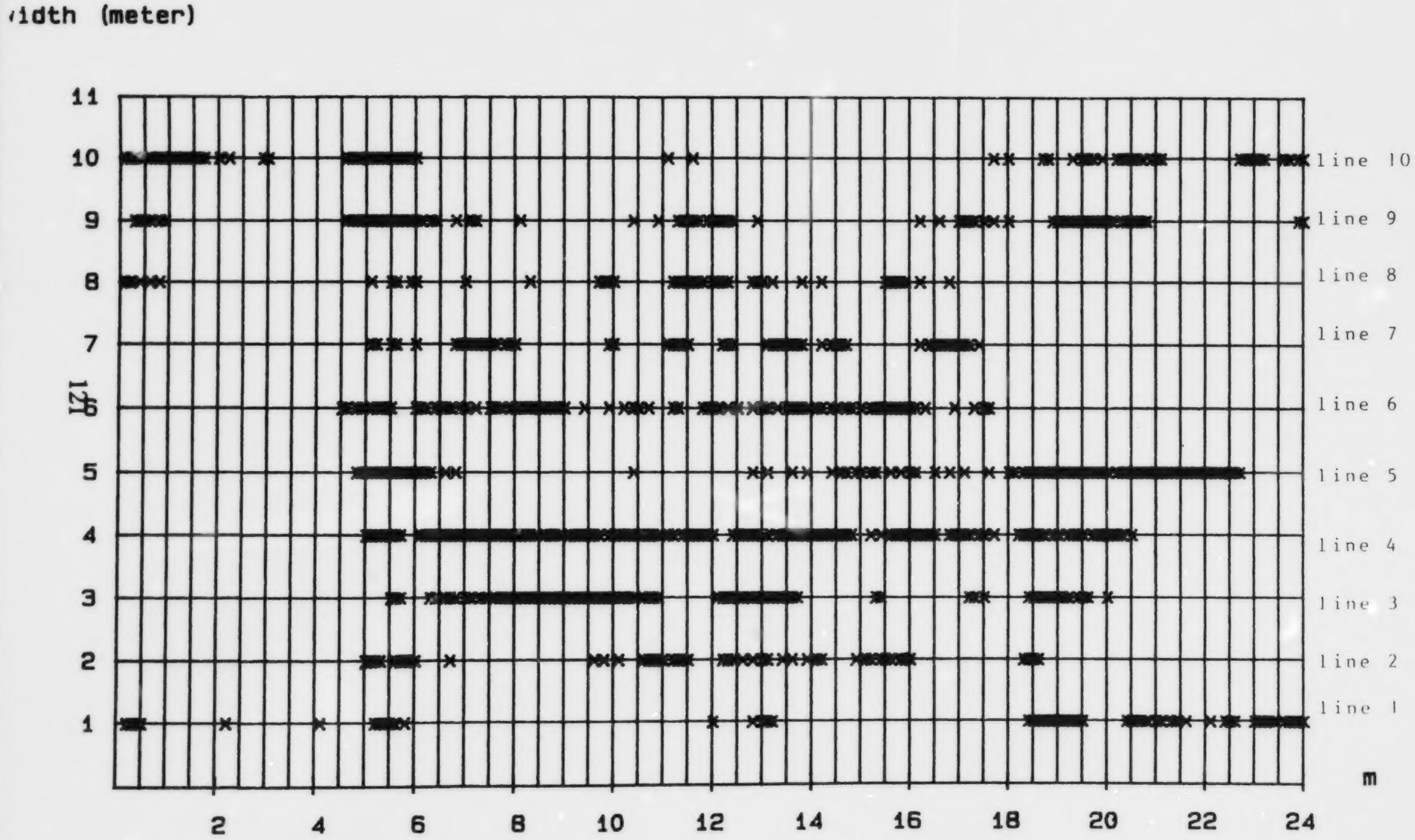
LITTLE BOB: SOUTH BRIDGE site 32-079, NB. SCALING

Figure 56: Scaled Areas at South Bridge Detected by 386-DART.



LITTLE BOB: SOUTH BRIDGE site 32-079, NB, DEBONDING

Figure 57: Debonded Areas at South Bridge Detected by 386-DART.



LITTLE BOB: SOUTH BRIDGE site 32-079, NB, DELAMINATION

Figure 58: Delaminaed Areas at South Bridge Detected by 386-DART.

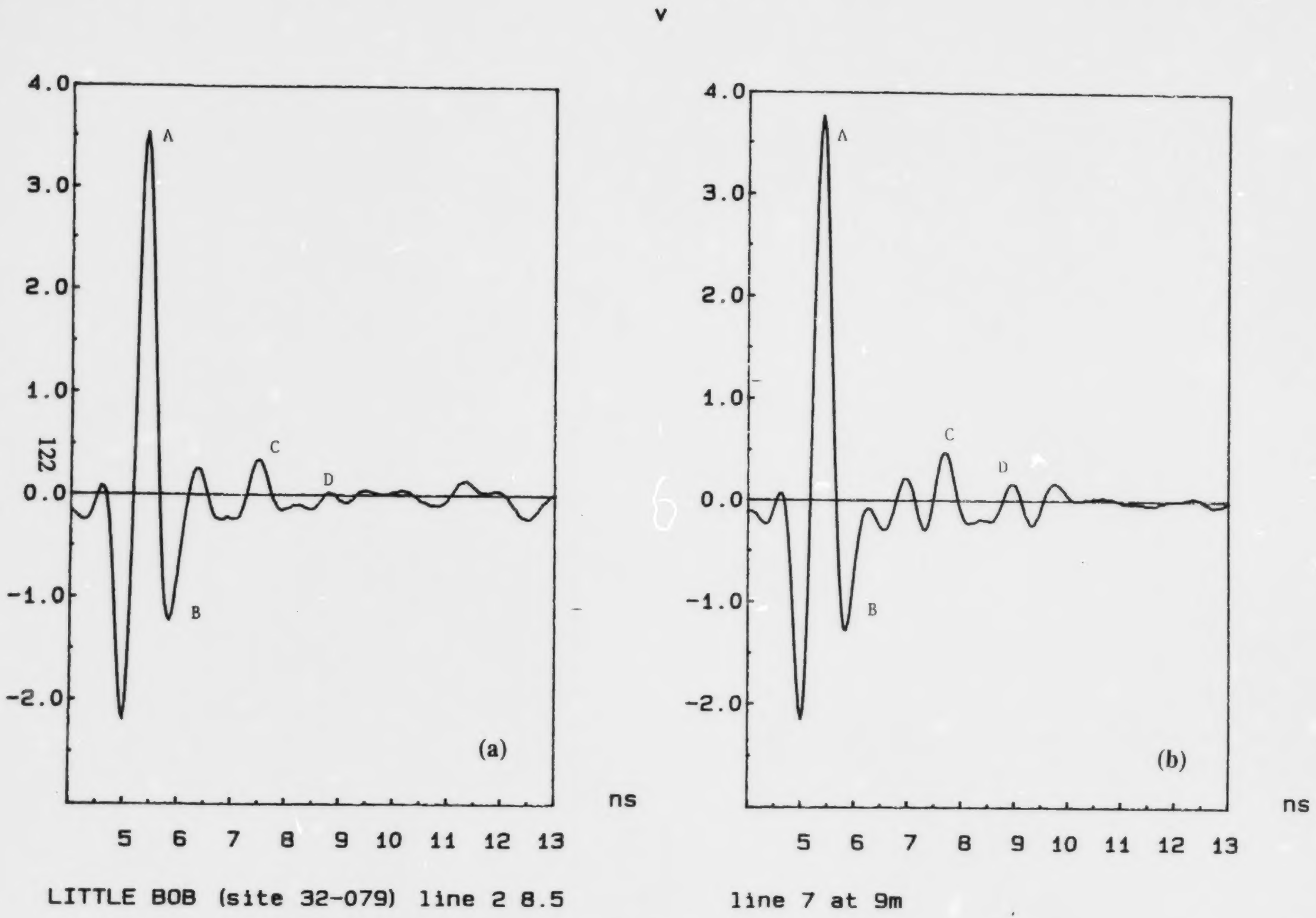


Figure 59: Little Bob South Bridge Waveforms Indicate Good Condition
(a) Line 2 at 8.5m, (b) Line 7 at 9m.

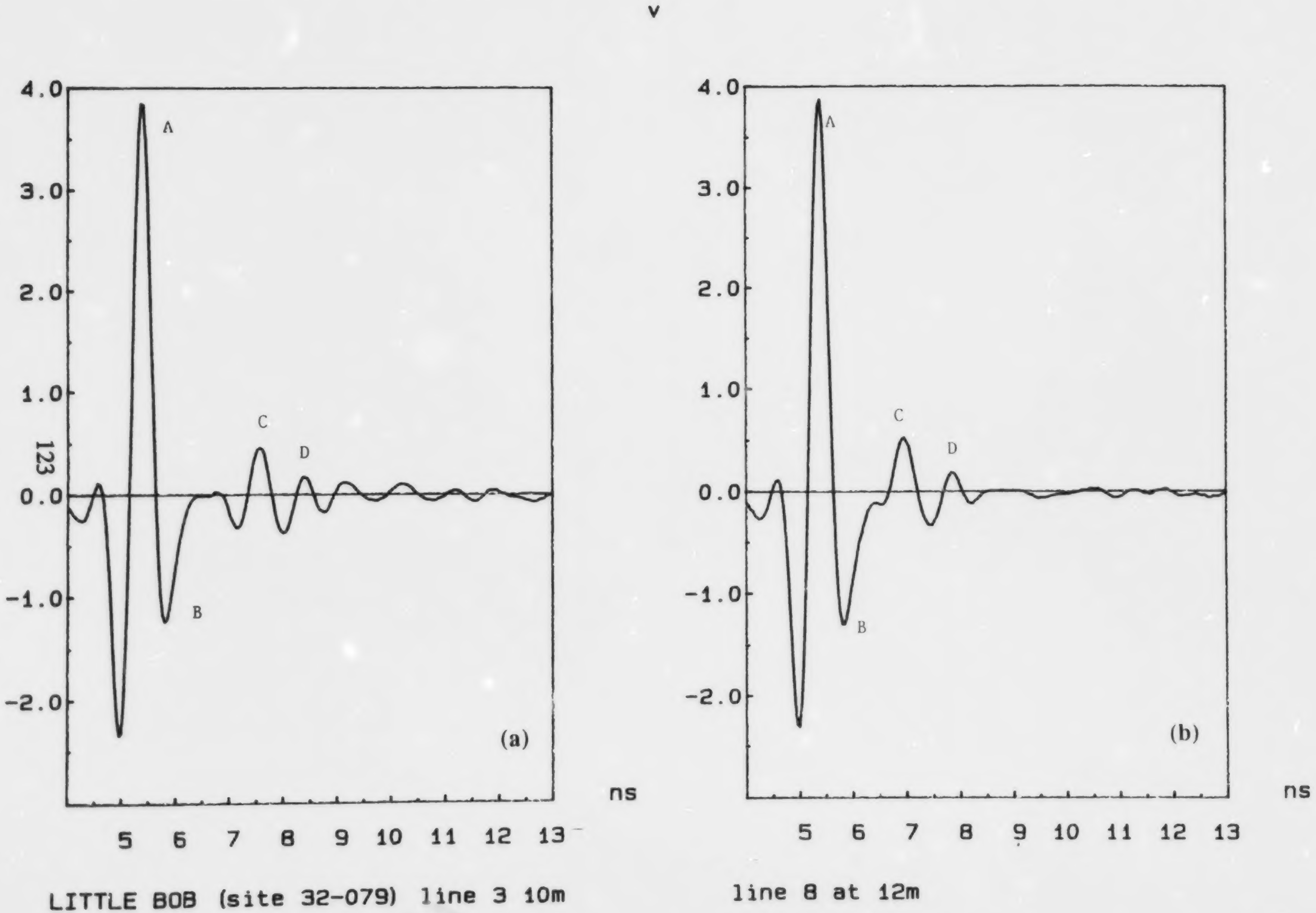


Figure 60: Little Bob South Bridge Waveforms Indicate Delamination
(a) Line 3 at 10m, (b) Line 8 at 12m.

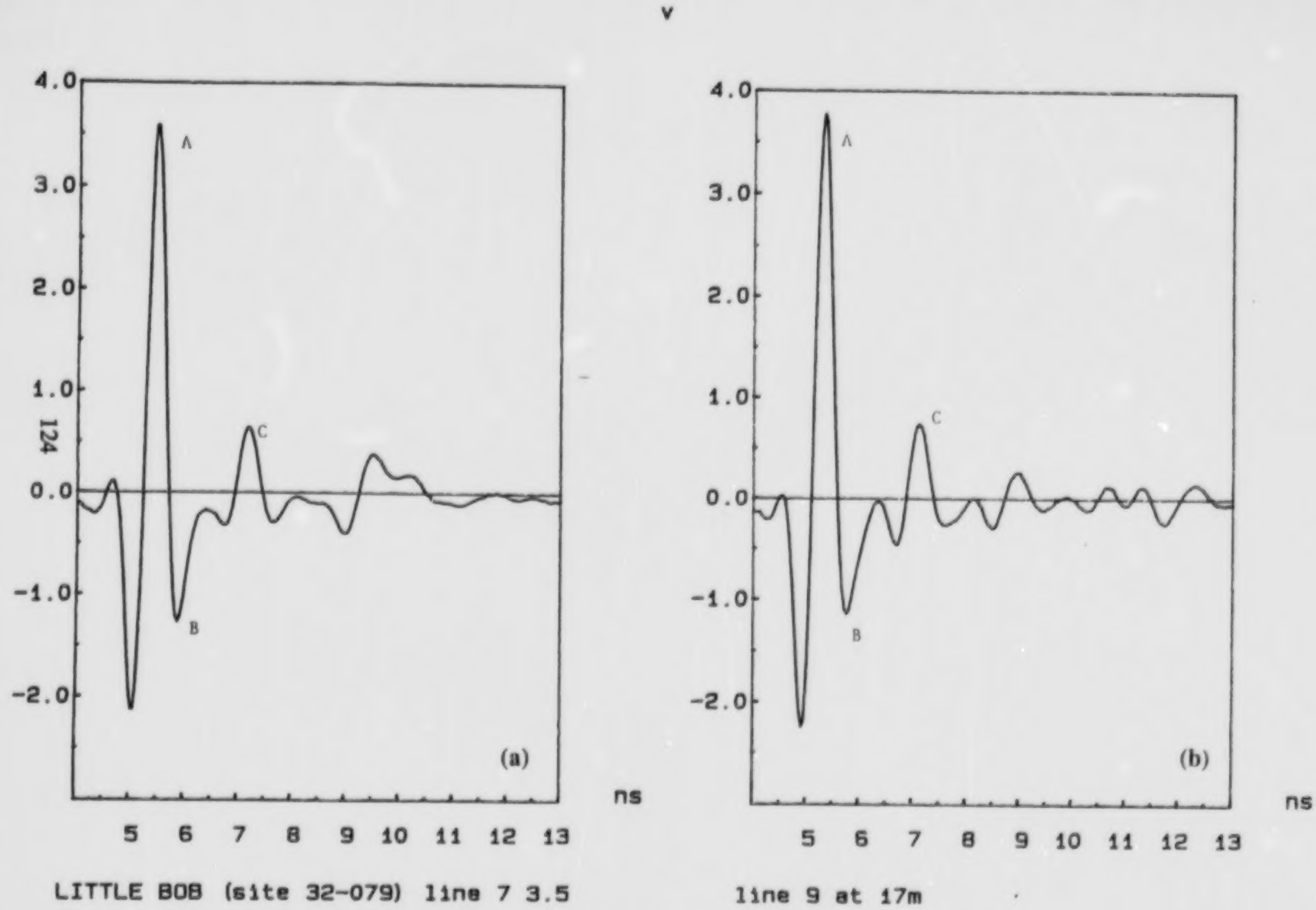


Figure 61: Little Bob South Bridge Waveforms Indicate Debonding
(a) Line 7 at 3.5m, (b) line 9 at 17m.

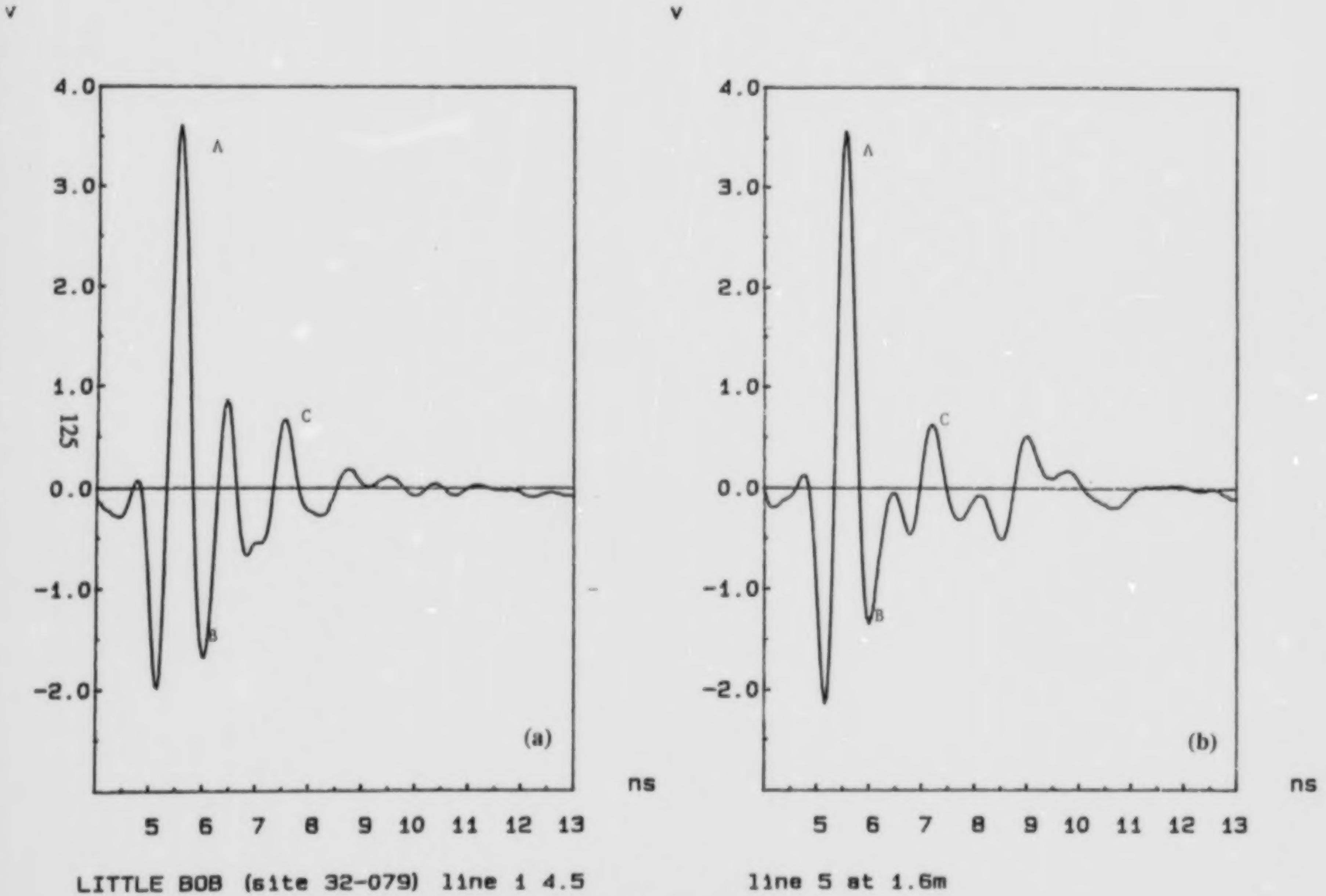
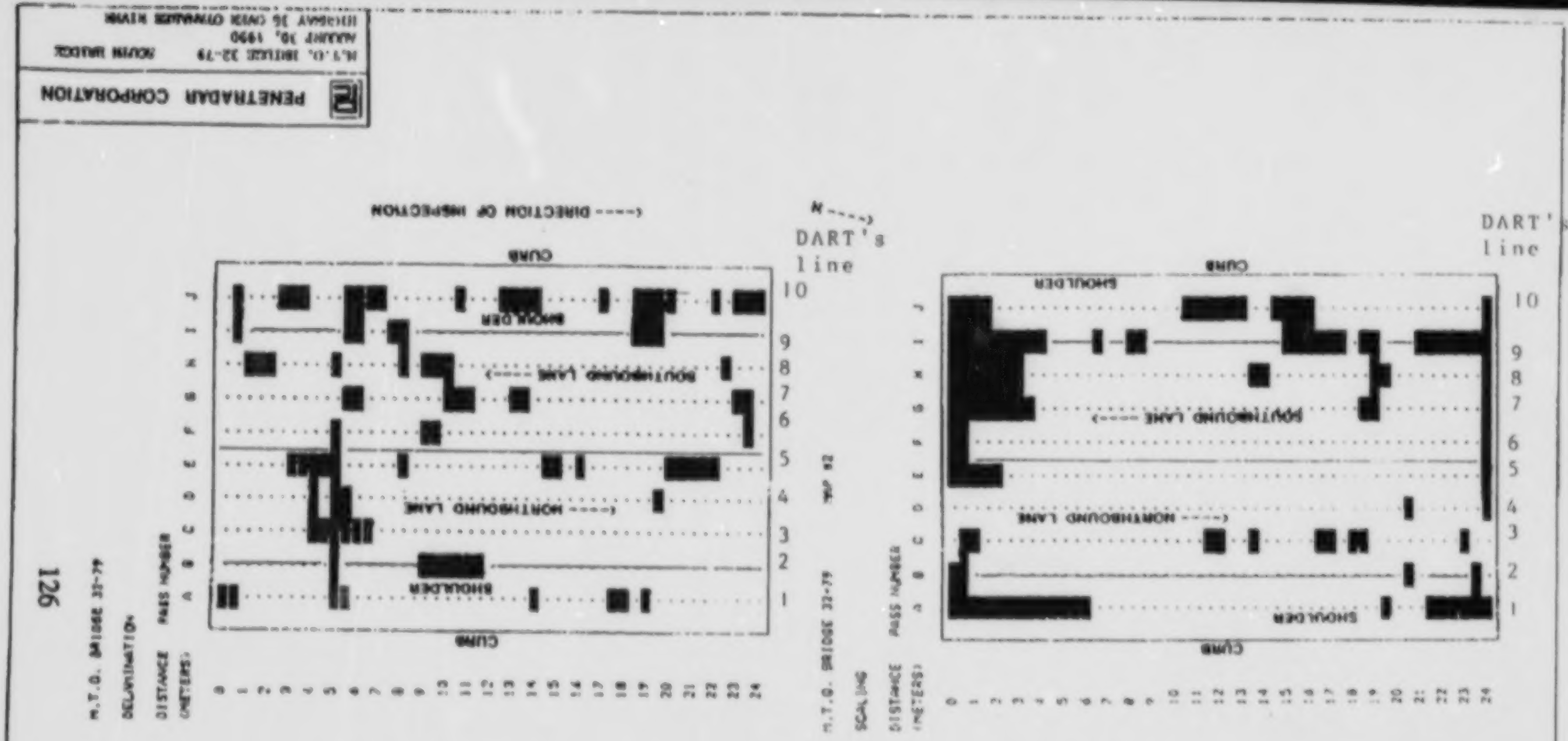


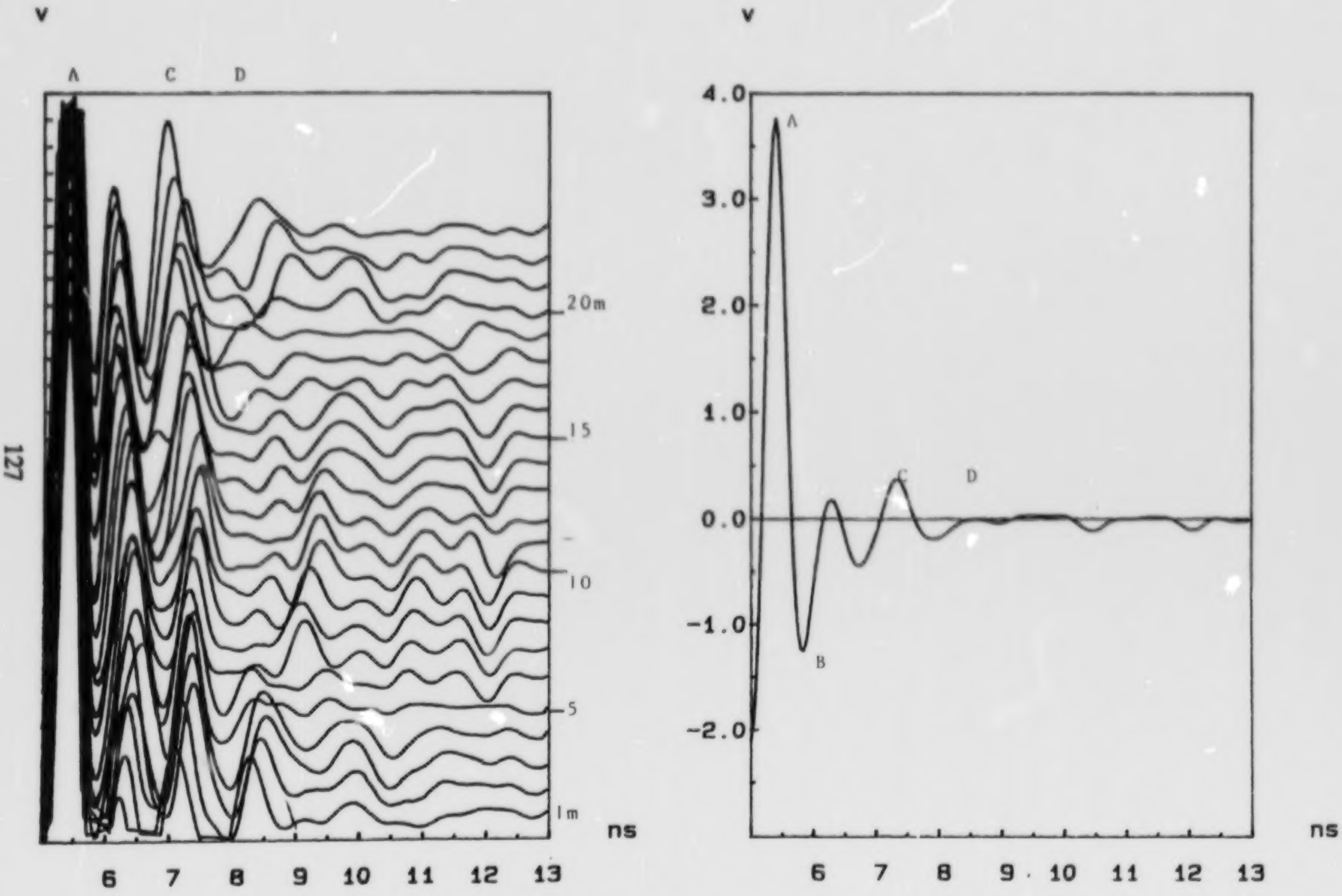
Figure 62: Little Bob South Bridge Waveforms Indicate Scaling
(a) Line 1 at 4.5m, (b) Line 5 at 1.6m.



Delamination

Scaling

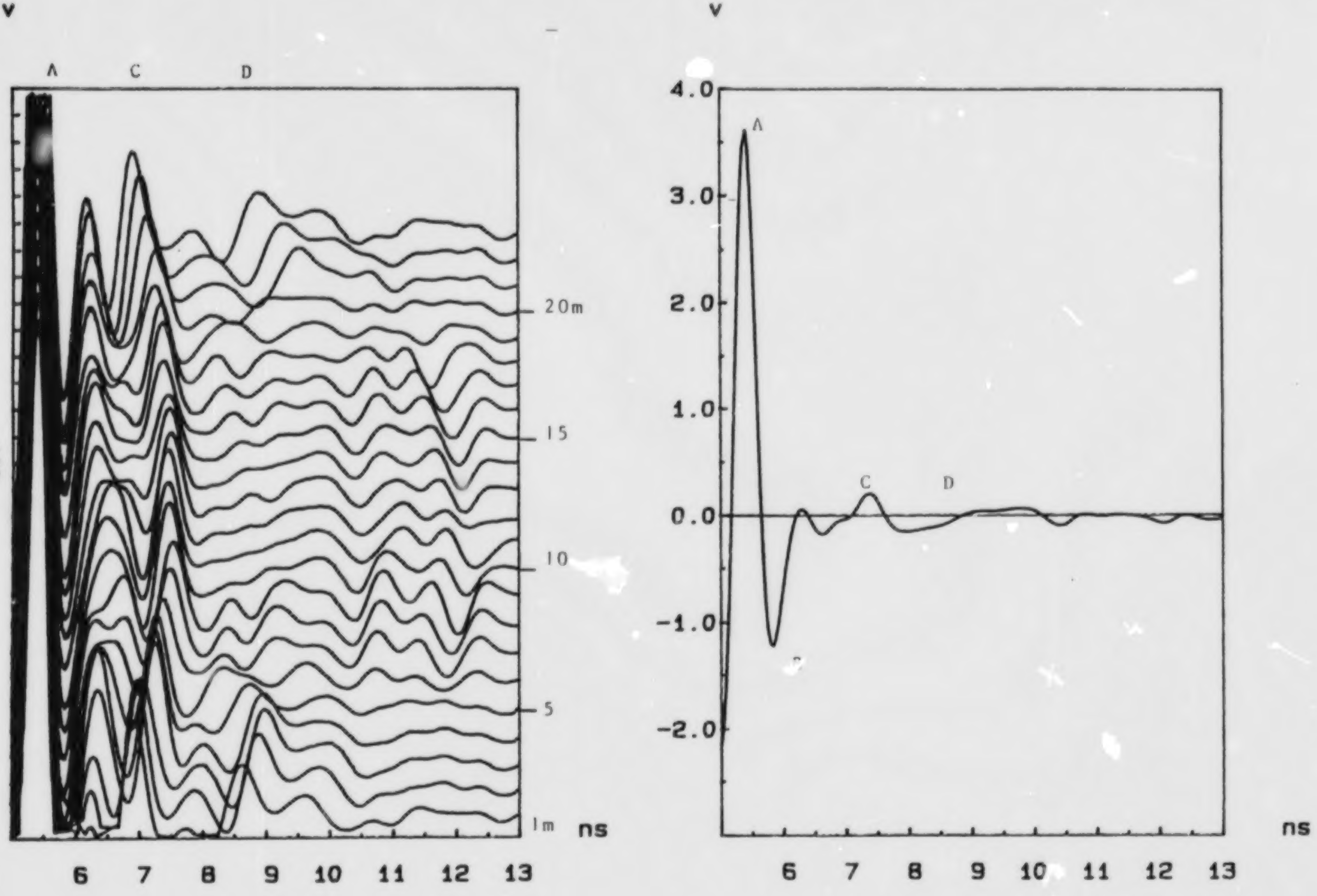
Figure 63: Delaminated and Scaled Areas at Little Bob South Bridge Detected by Penetradar.



Little Bob, site 32-080, line 1

average waveform

Figure 64: Waveforms from Line 1 of Little Bob North Bridge.



Little Bob. site 32-080. line 2

average waveform

Figure 65: Waveforms from Line 2 of Little Bob North Bridge.

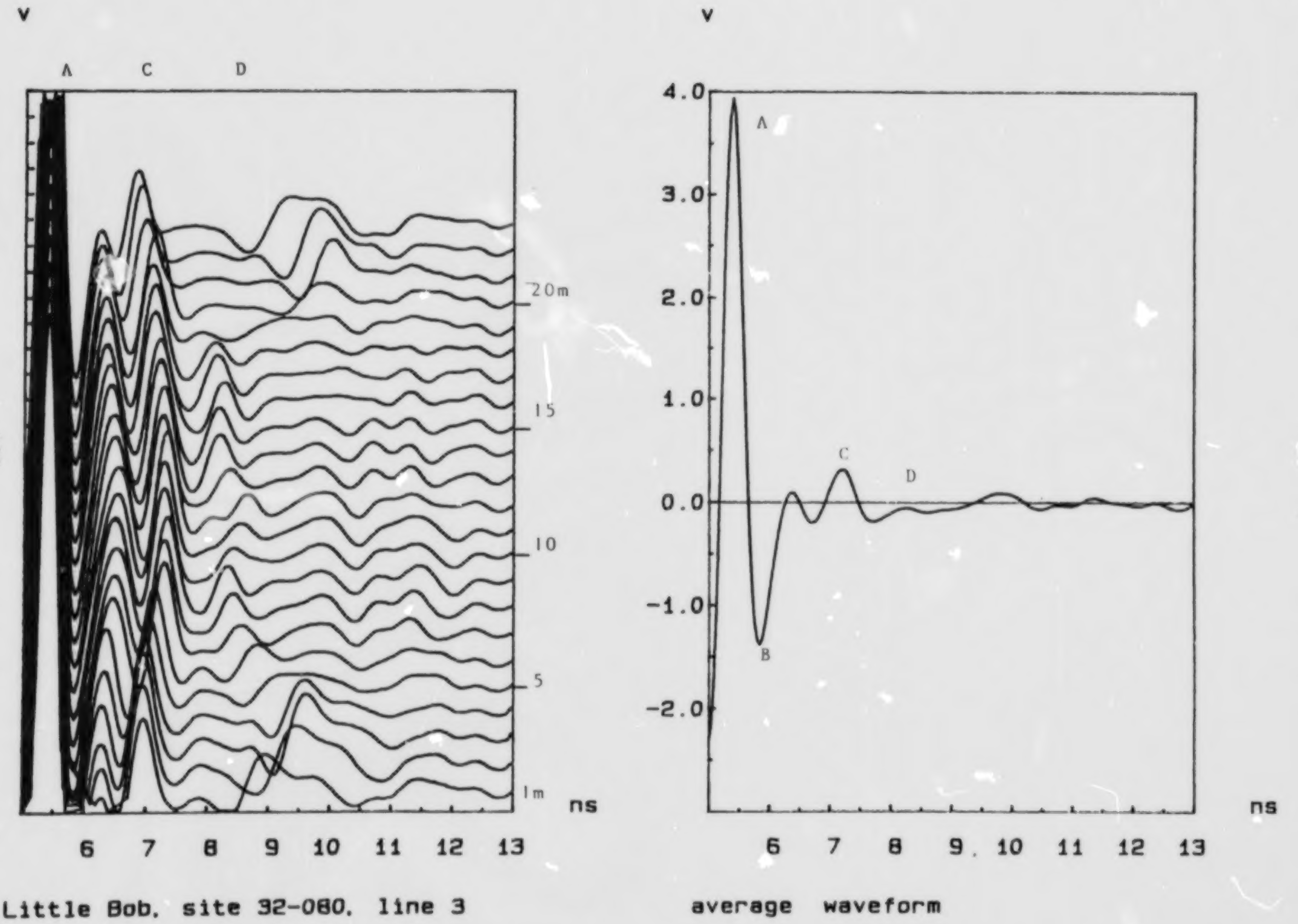
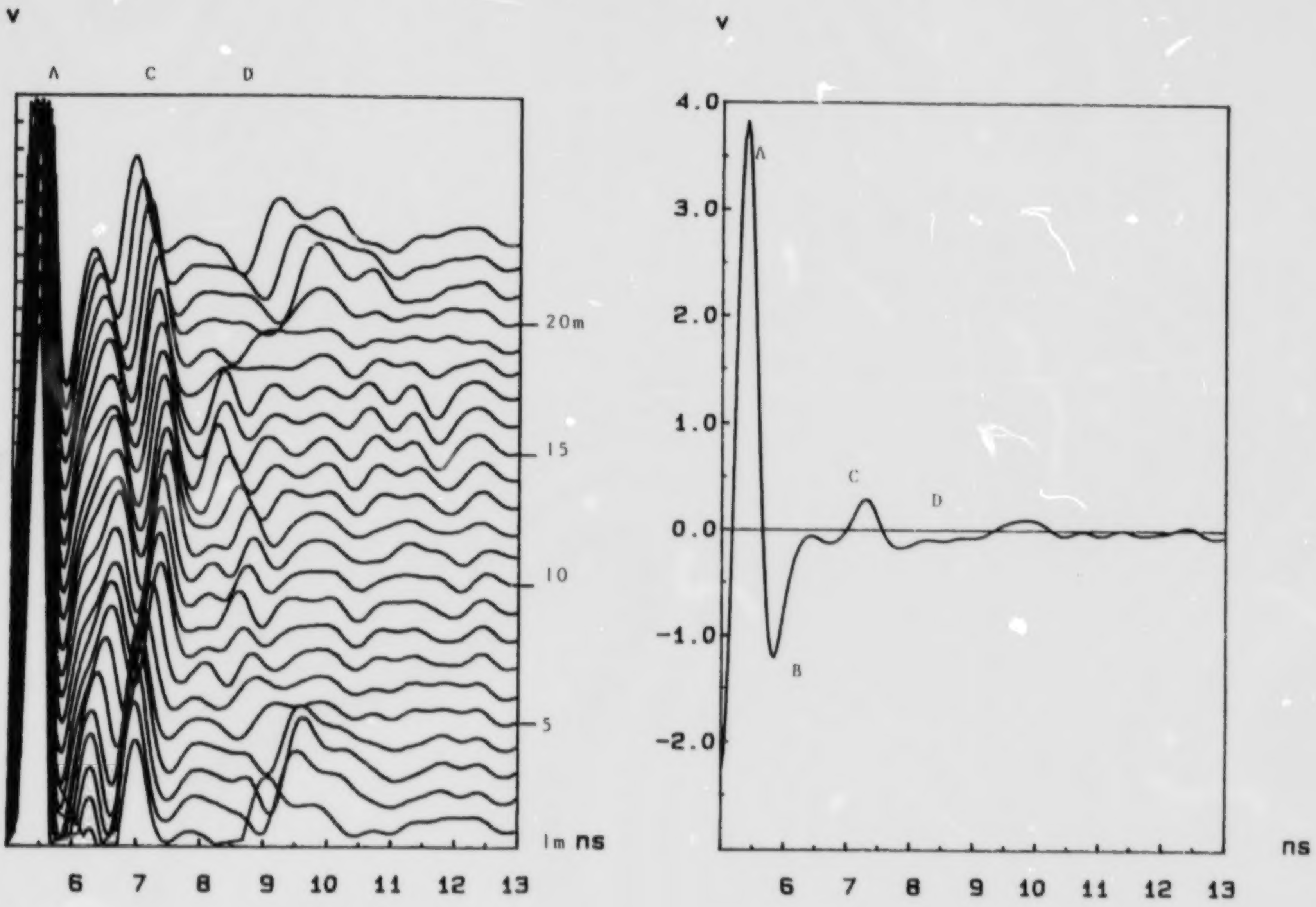


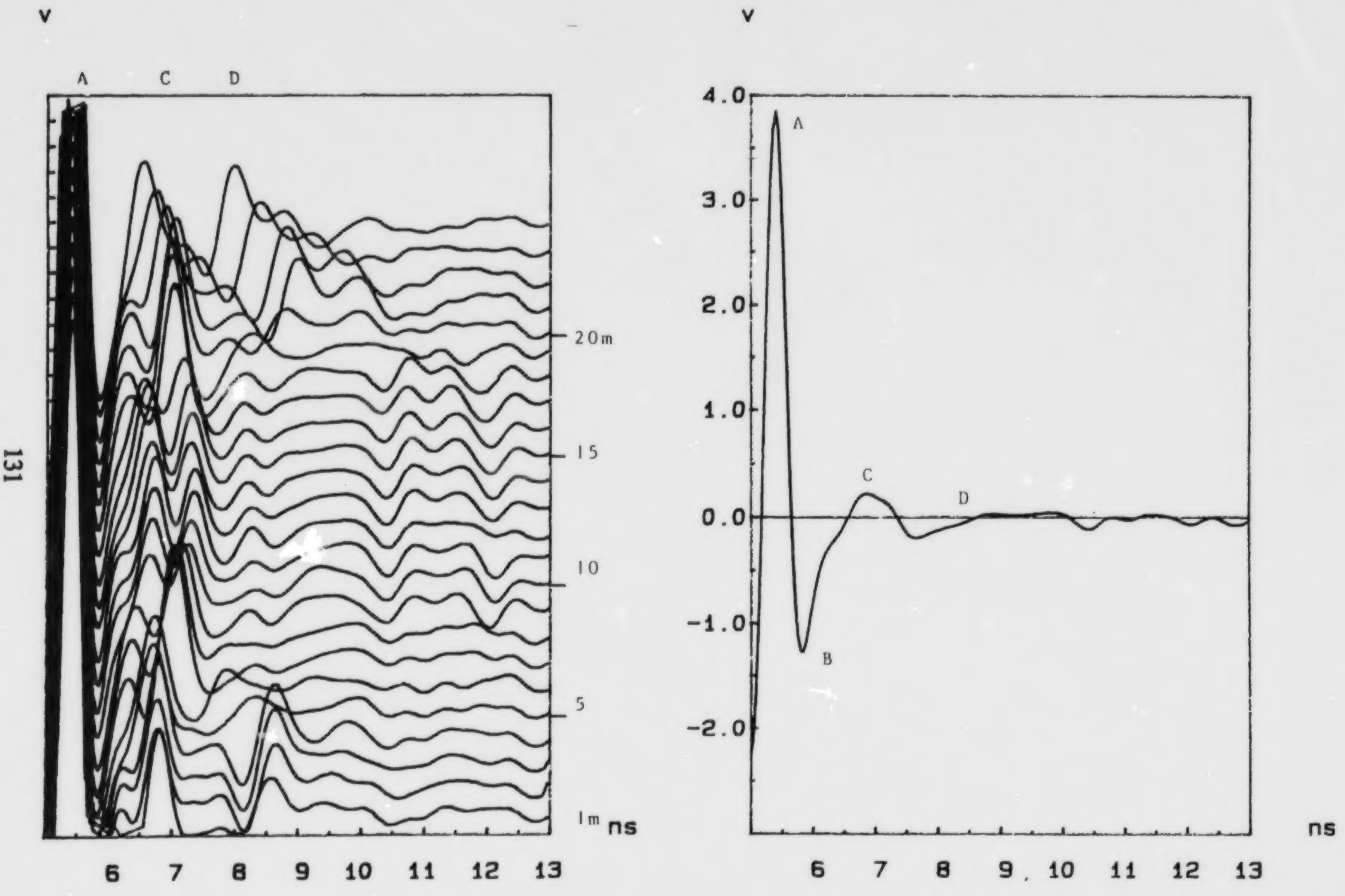
Figure 66: Waveforms from Line 3 of Little Bob North Bridge.



Little Bob. site 32-080. line 4

average waveform

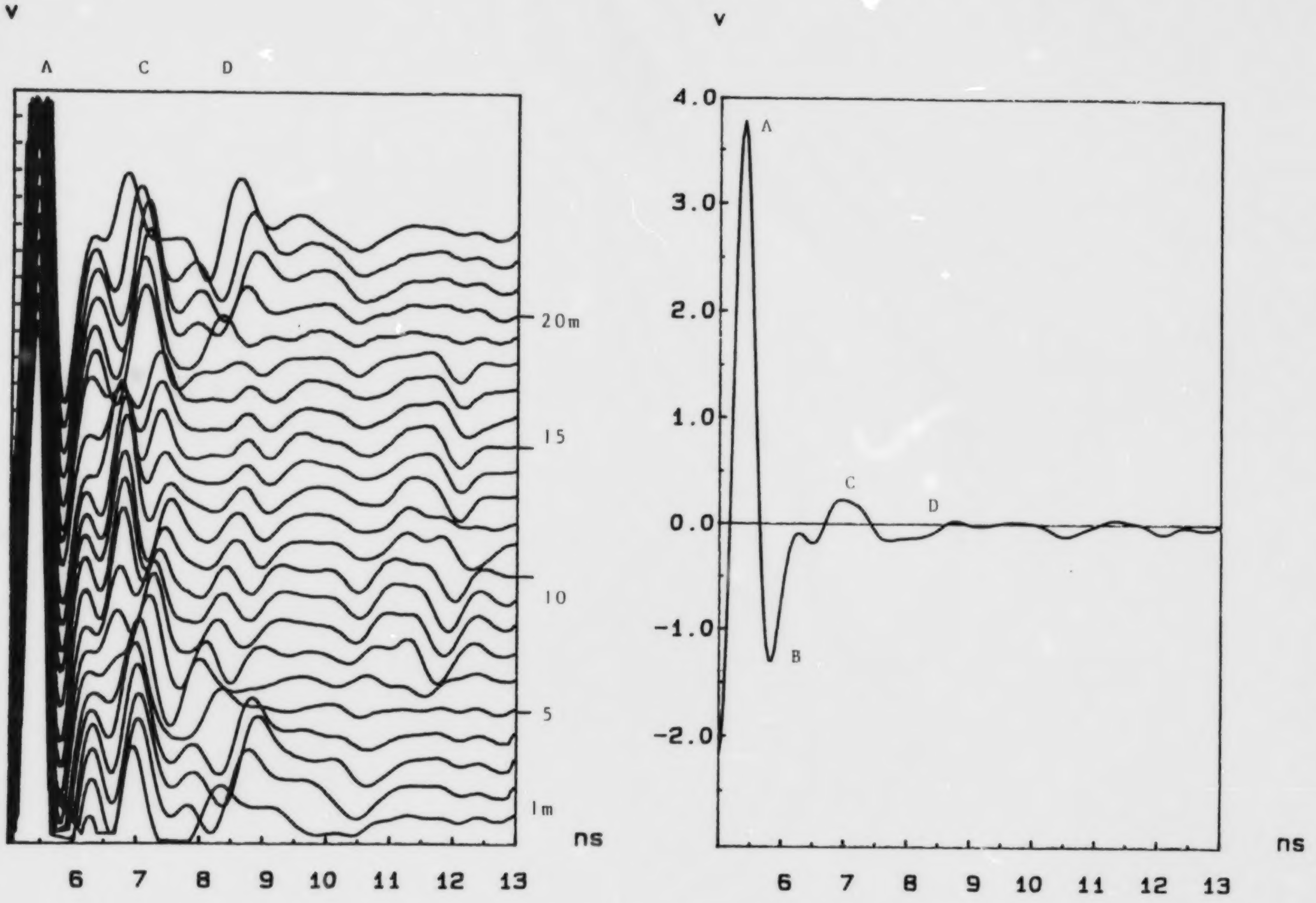
Figure 67: Waveforms from Line 4 of Little Bob North Bridge.



Little Bob. site 32-080. line 5

average waveform

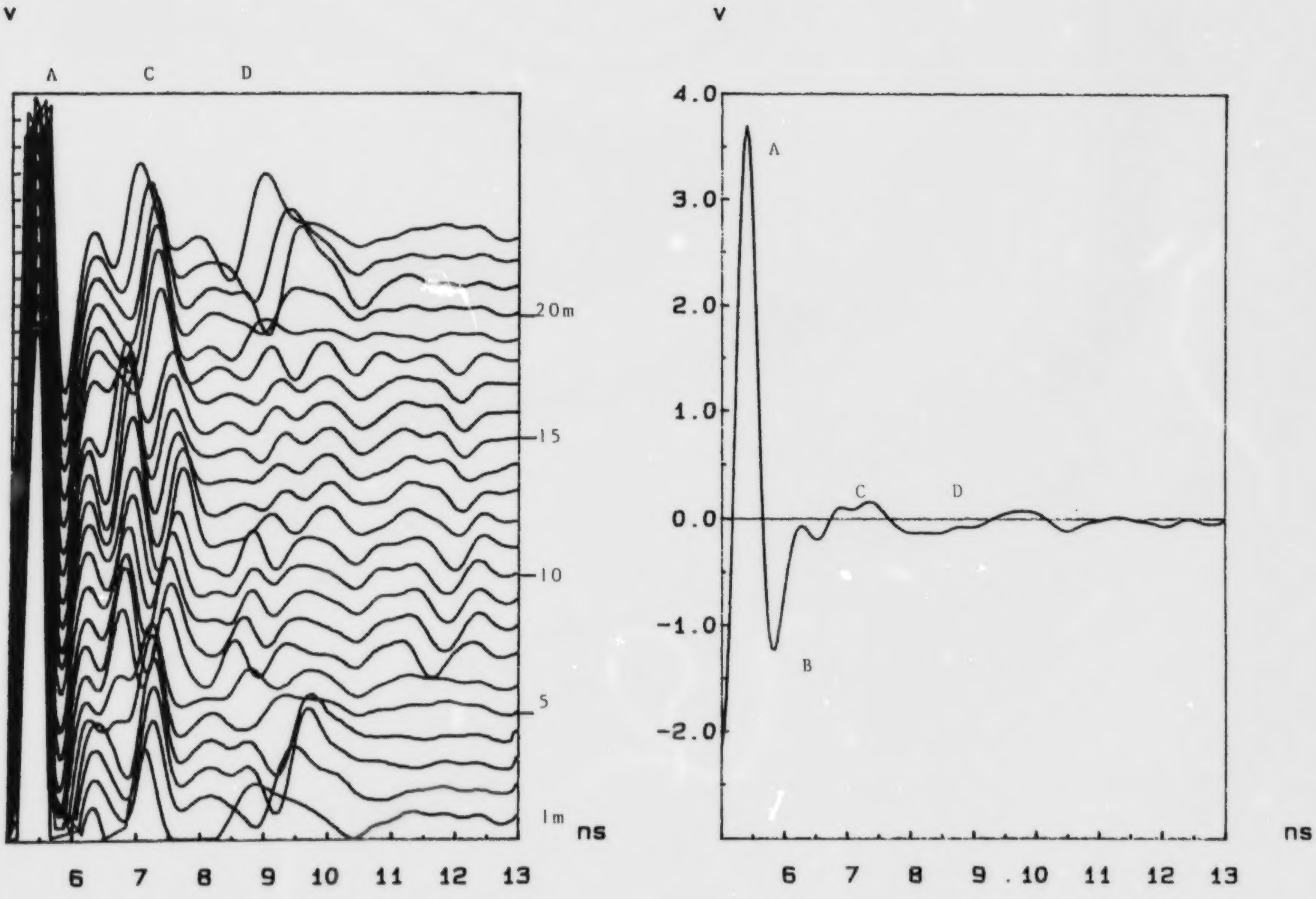
Figure 68: Waveforms from Line 5 of Little Bob North Bridge.



Little Bob, site 32-080, line 6

average waveform

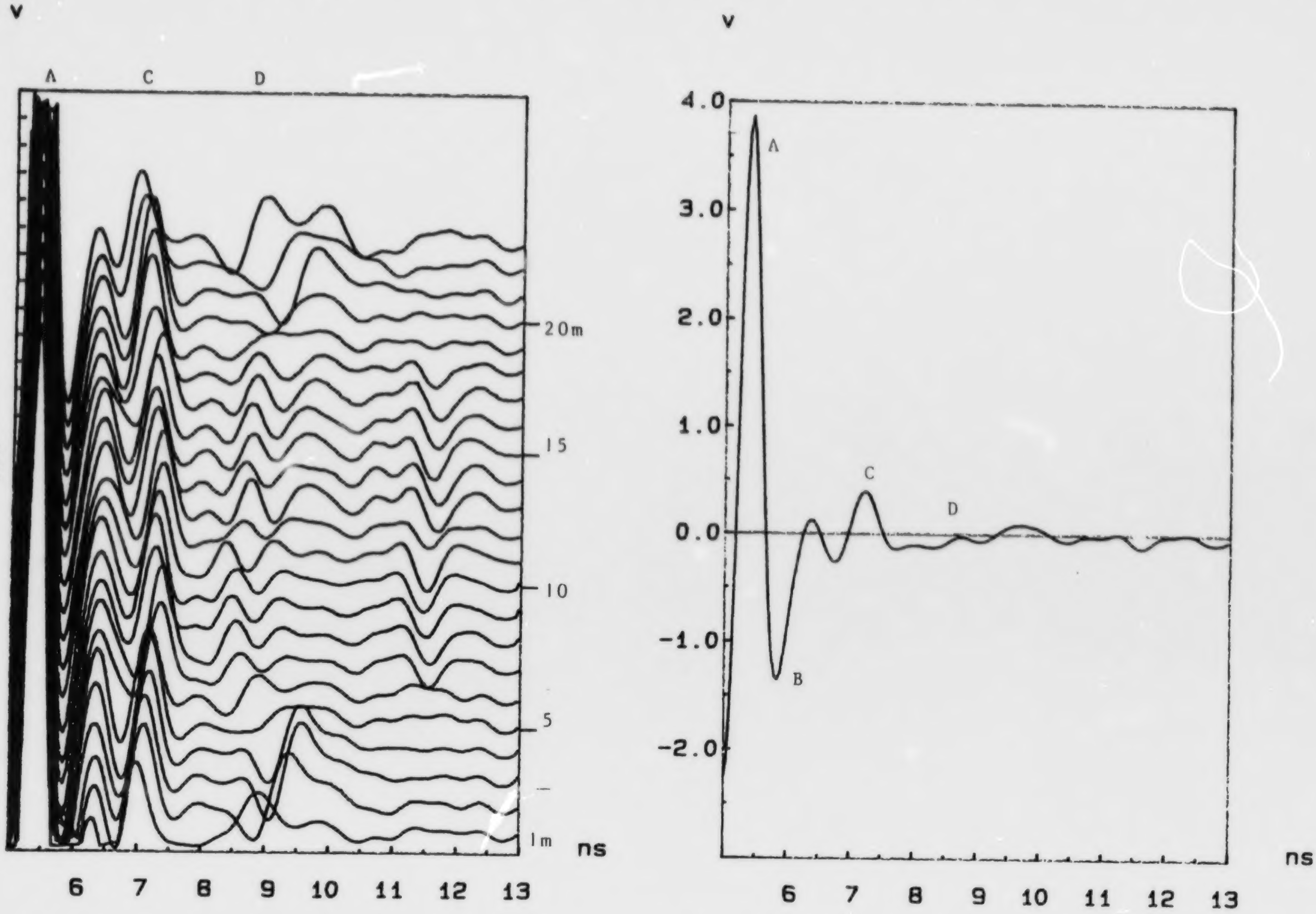
Figure 69: Waveforms from Line 6 of Little Bob North Bridge.



Little Bob. site 32-080. line 7

average waveform

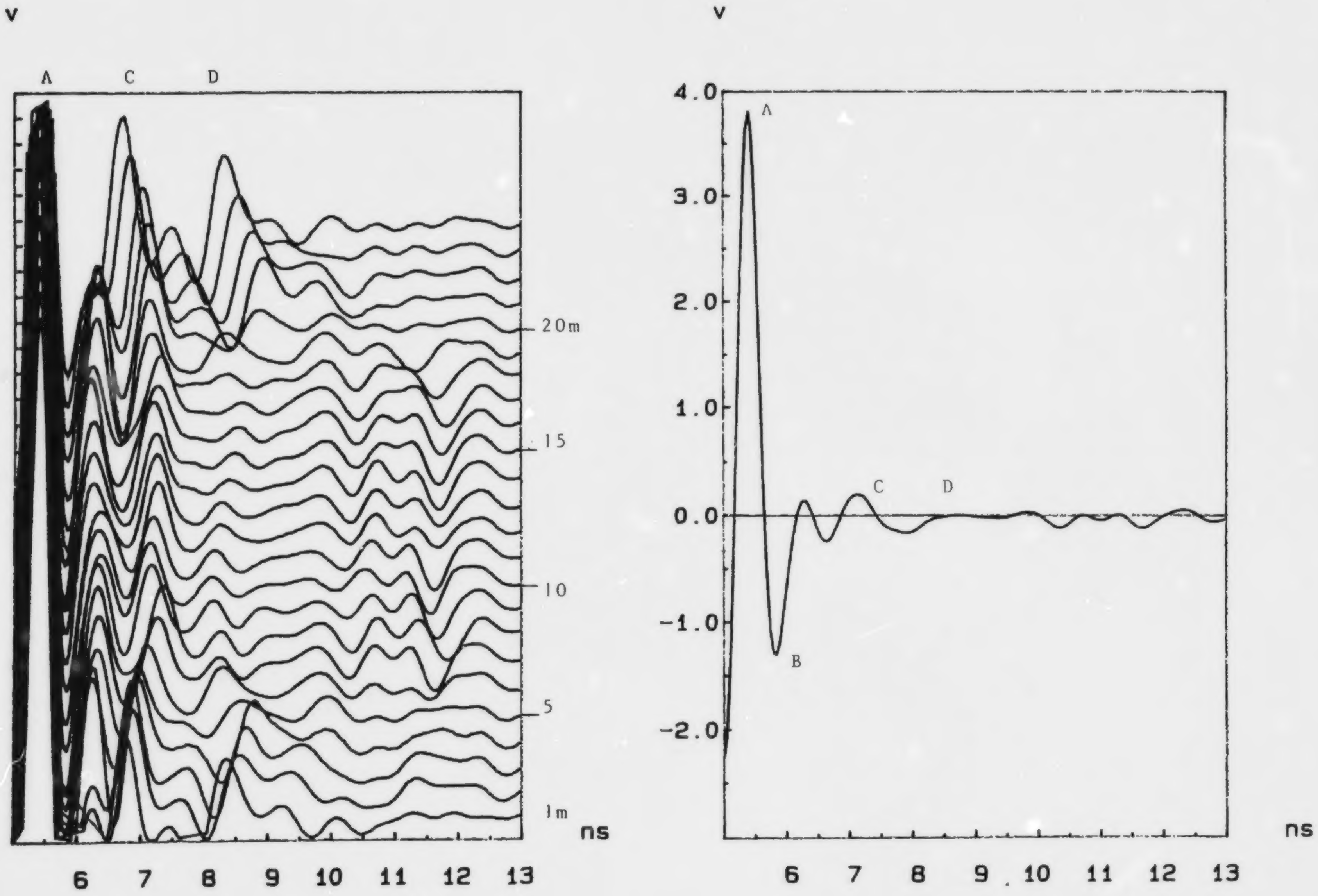
Figure 70: Waveforms from Line 7 of Little Bob North Bridge.



Little Bob, site 32-080, line 8

average waveform

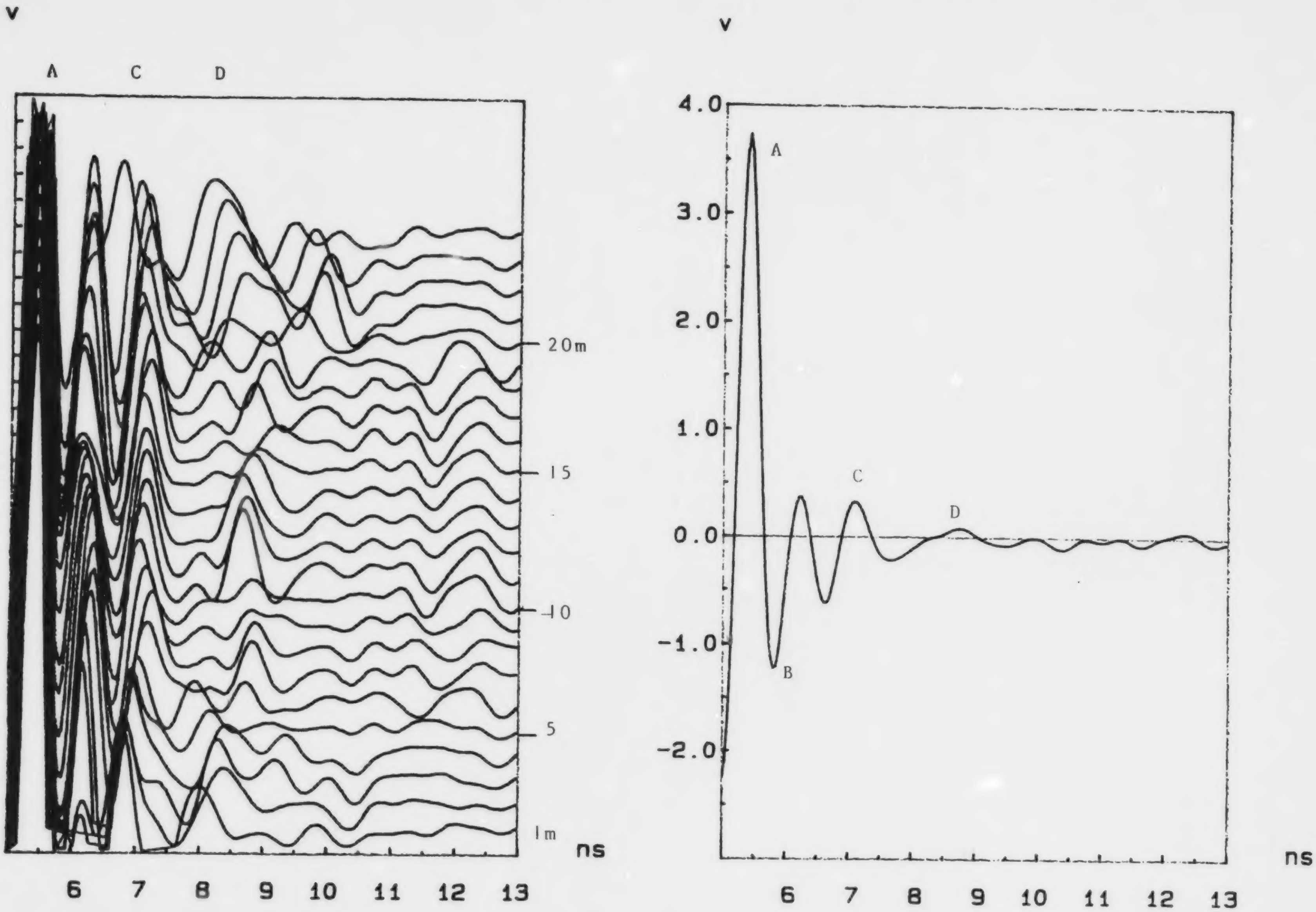
Figure 71: Waveforms from Line 8 of Little Bob North Bridge.



Little Bob. site 32-080. line 9

average waveform

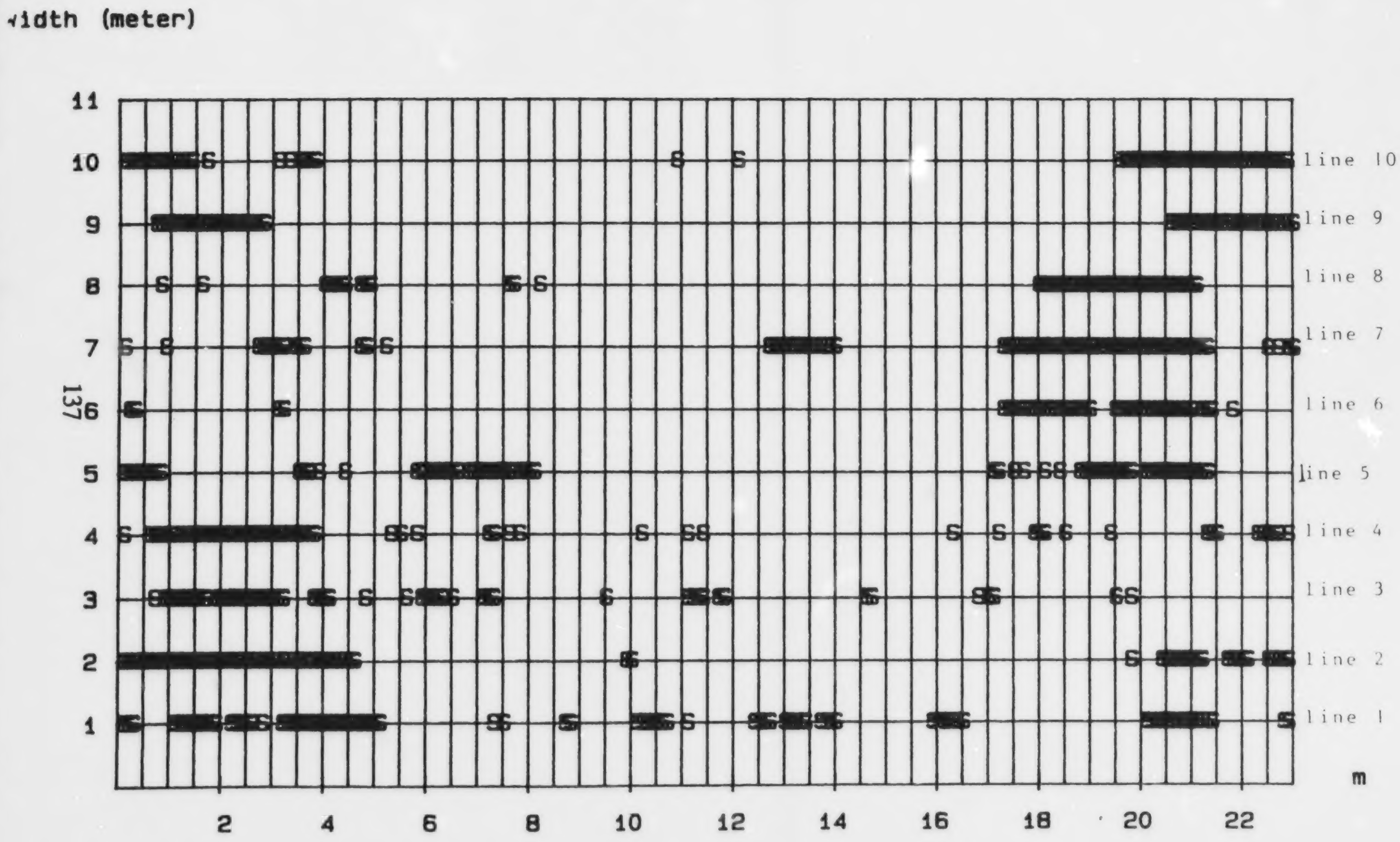
Figure 72: Waveforms from Line 9 of Little Bob North Bridge.



Little Bob. site 32-080. line 10

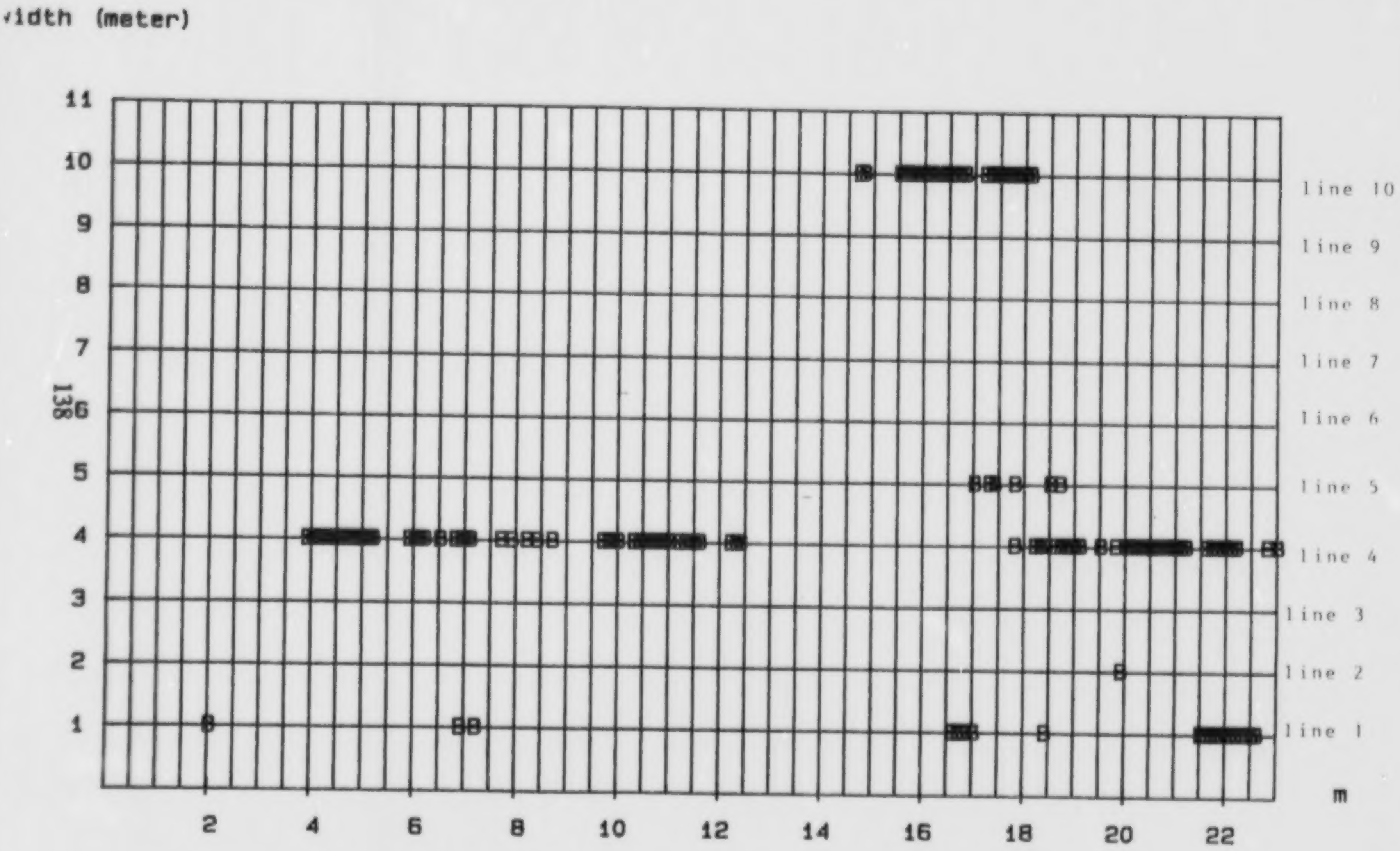
average waveform

Figure 73: Waveforms from Line 10 of Little Bob North Bridge.



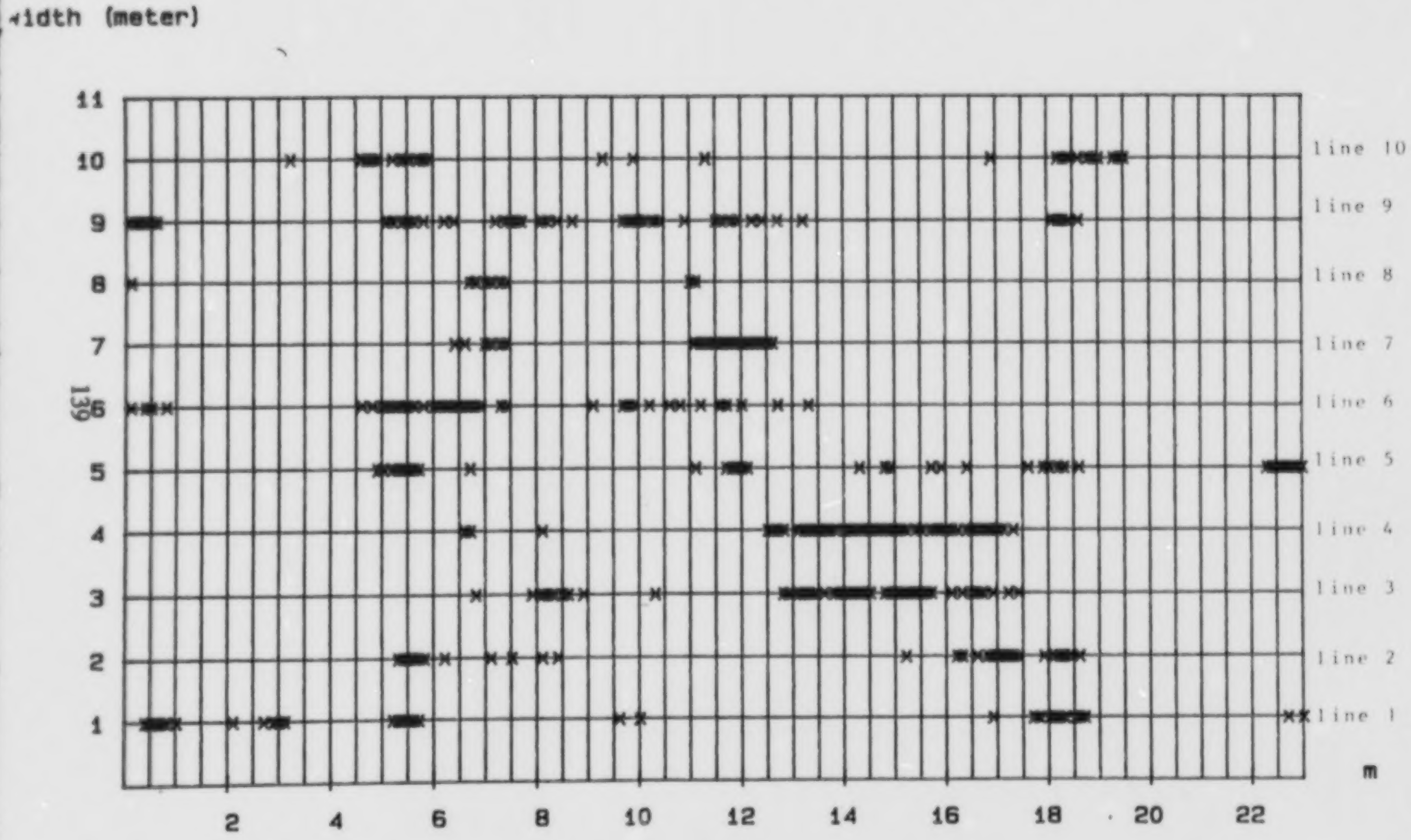
LITTLE BOB NORTH BRIDGE: site 32-080, NB, SCALING

Figure 74: Scaled Areas at North Bridge Detected by 386-DART.



LITTLE BOB NORTH BRIDGE: site 32-080, NB, DEBONDING

Figure 75: Debonded Areas at North Bridge Detected by 386-DART.



LITTLE BOB NORTH BRIDGE: site 32-080, NB, DELAMINATION

Figure 76: Delaminated Areas at North Bridge Detected by 386-DART.

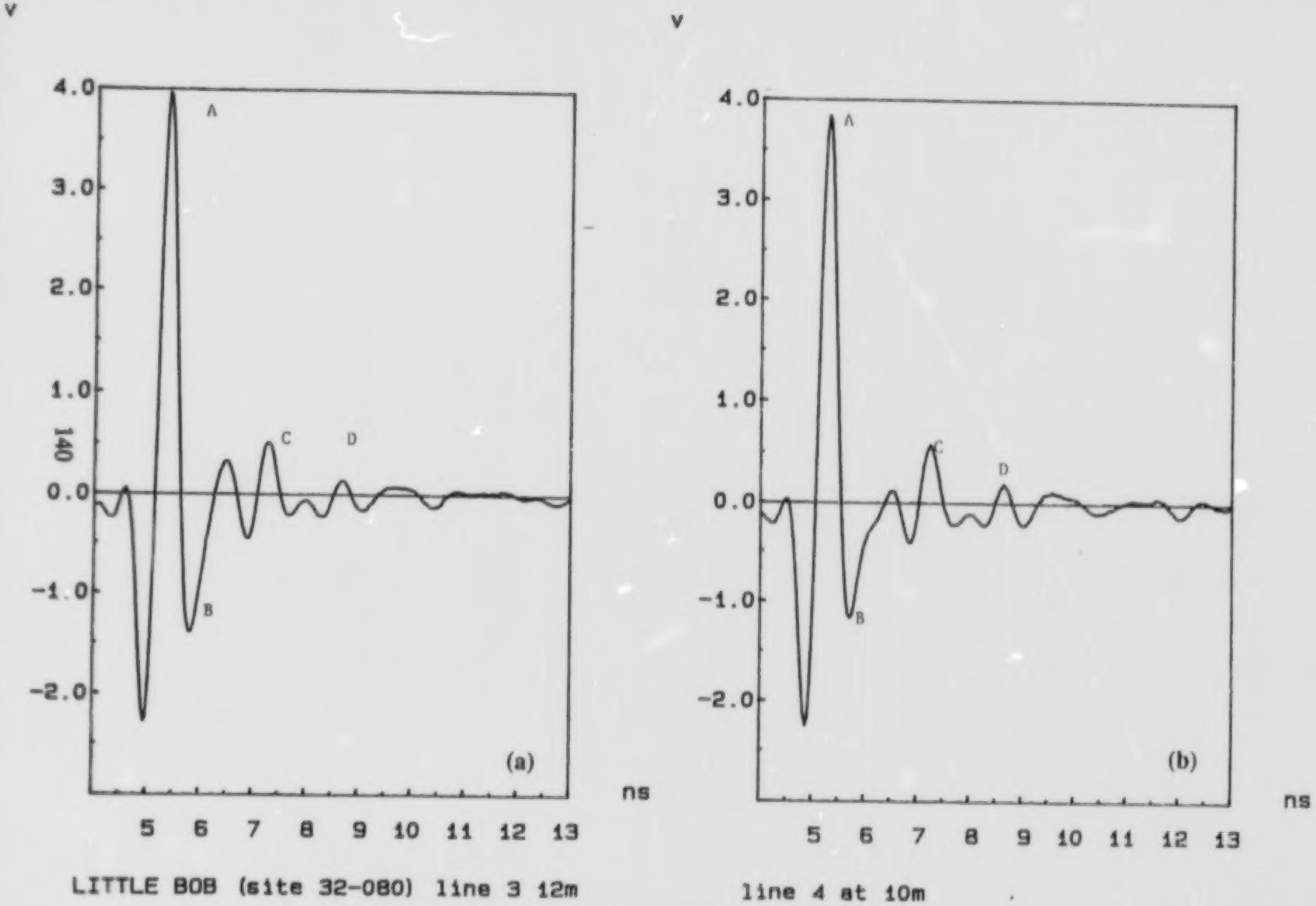


Figure 77: Little Bob North Bridge Waveforms Indicate Good Condition
(a) Line 2 at 12m. (b) Line 4 at 10m.

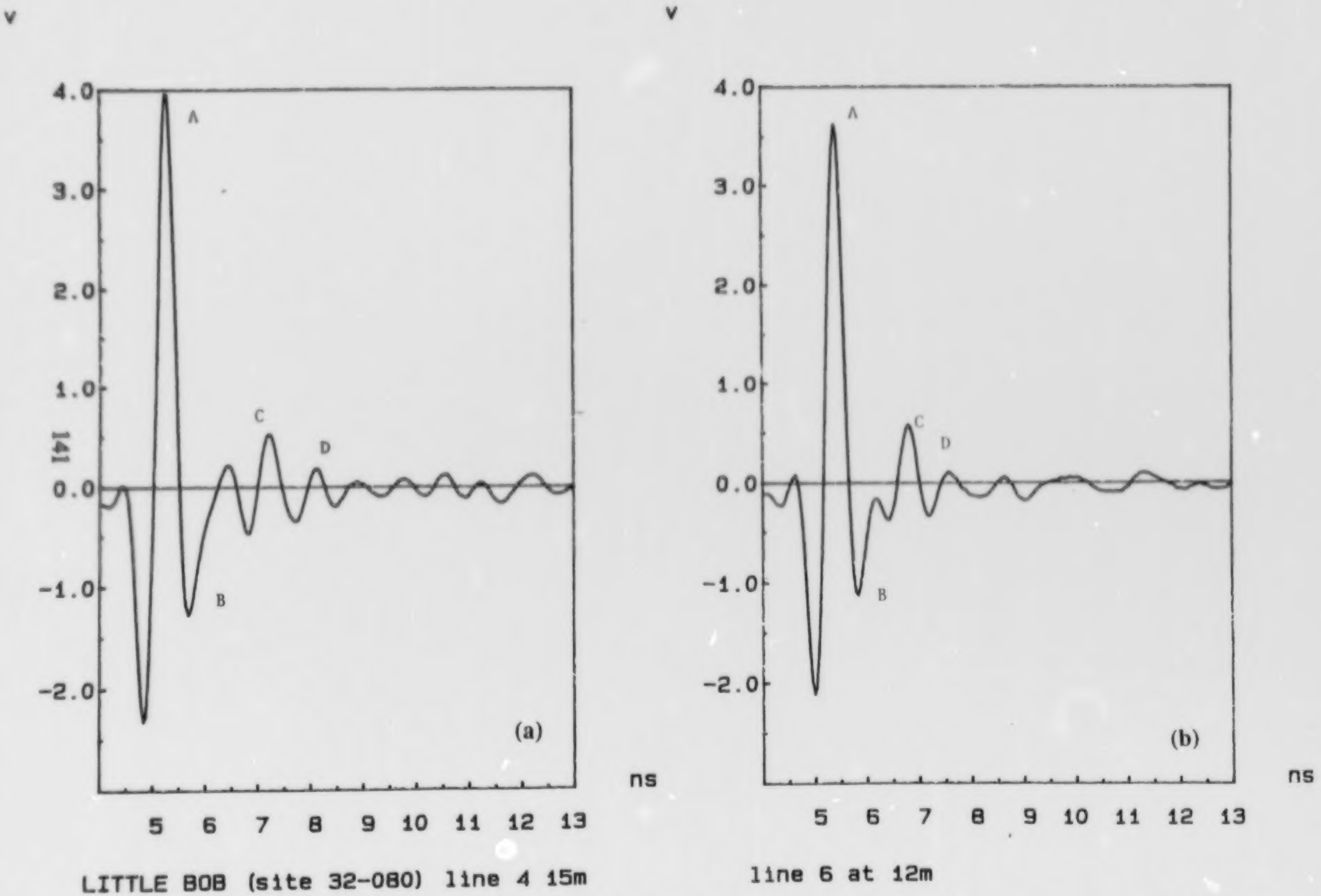


Figure 78: Little Bob North Bridge Waveforms Indicate Delamination
(a) Line 4 at 15m, (b) Line 6 at 12m.

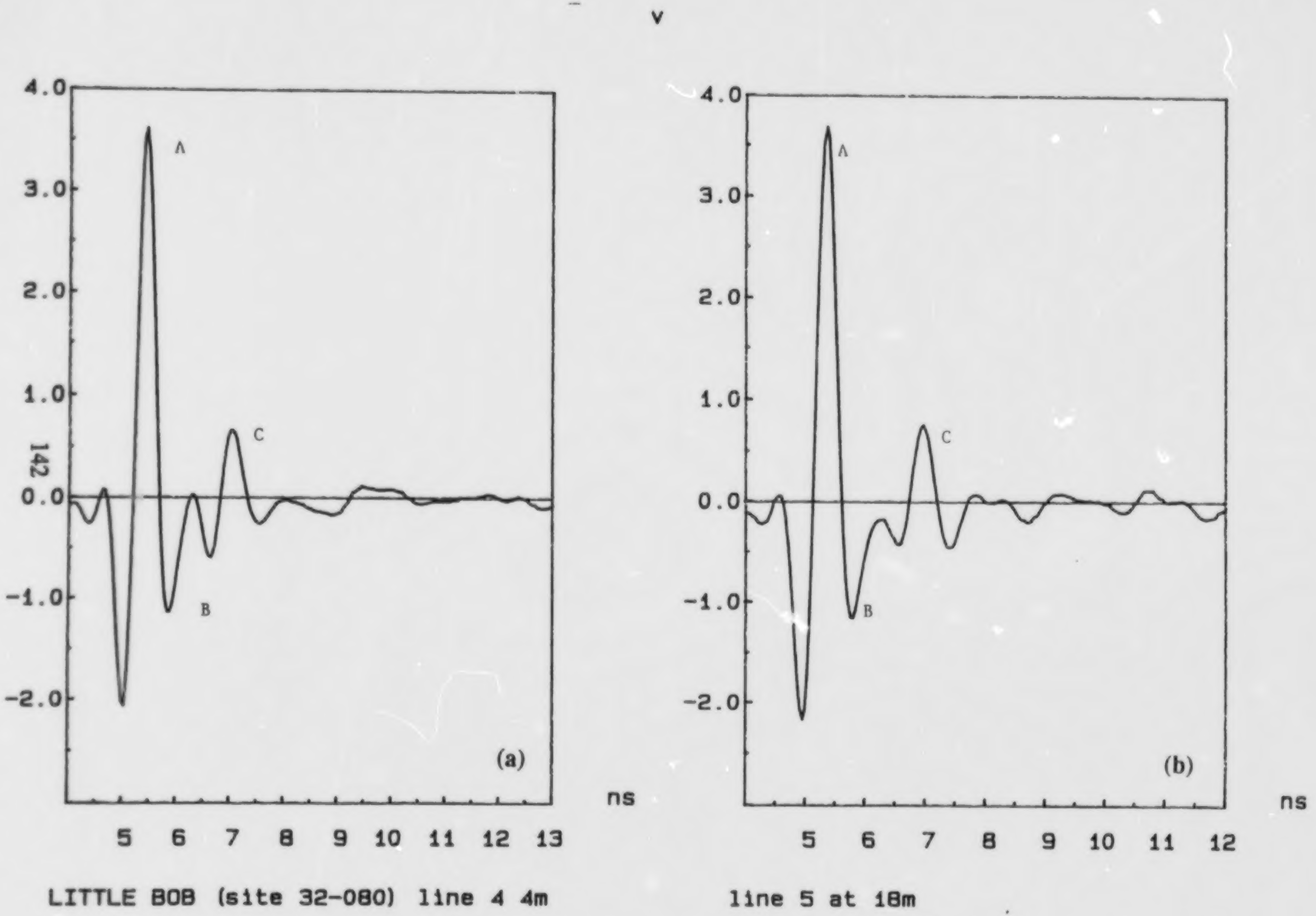


Figure 79: Little Bob North Bridge Waveforms Indicate Debonding
(a) Line 4 at 4m, (b) Line 5 at 18m.

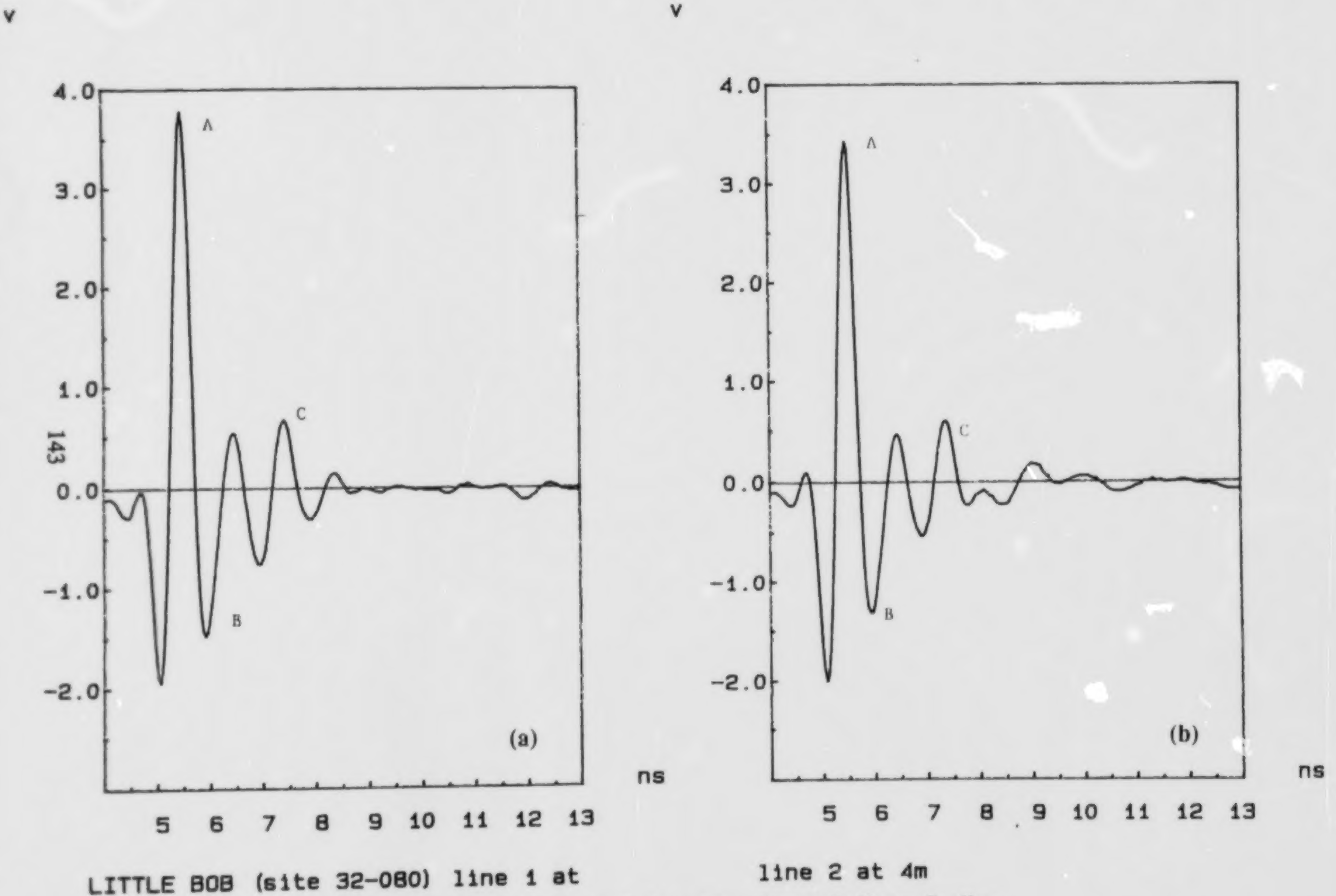
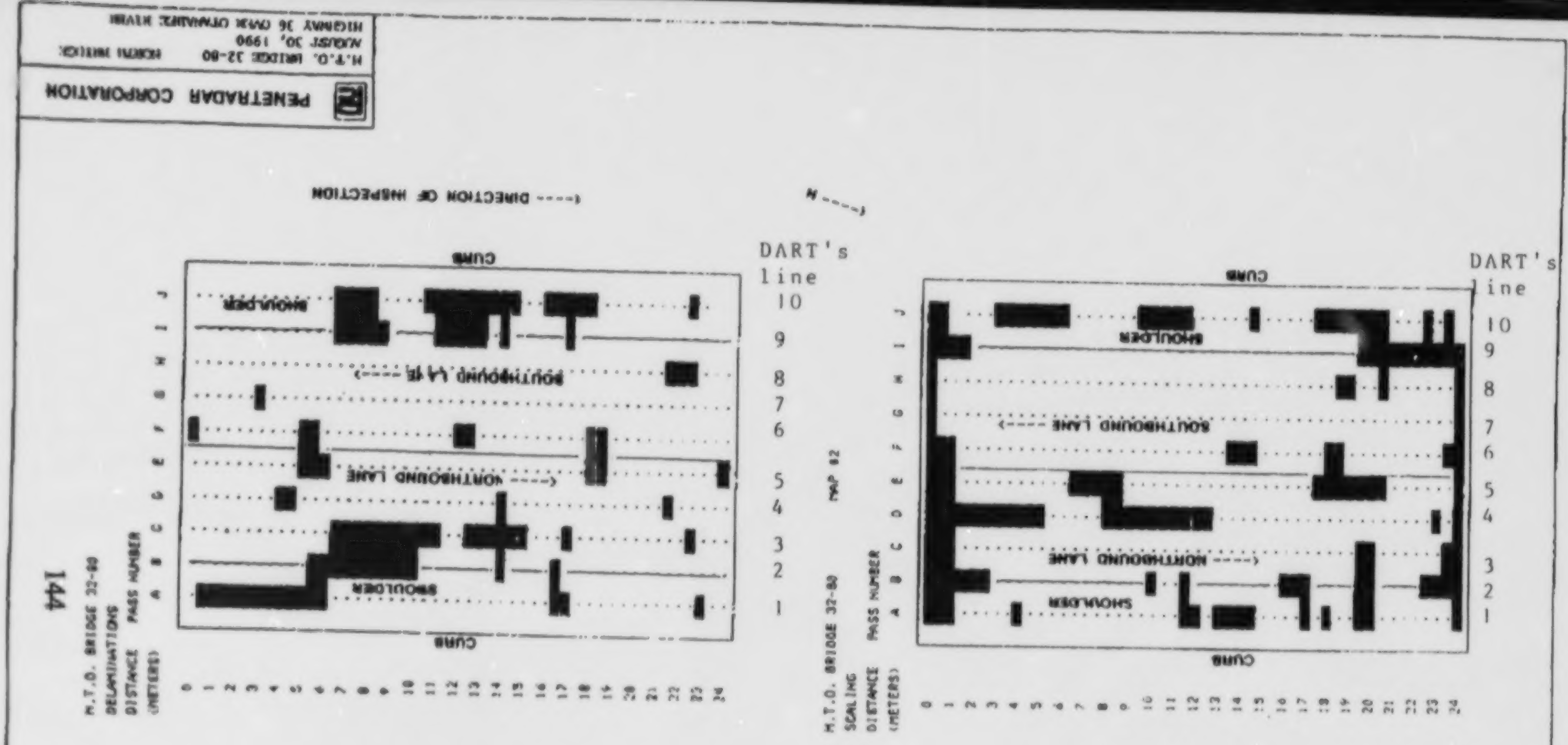


Figure 80: Little Bob North Bridge Waveforms Indicate Scaling
(a) Line 1 at 5m, (b) Line 2 at 4m.


PENETRADAR CORPORATION
M.T.O. BRIDGE 32-80
AUGUST 30, 1990
<---- DIRECTION OF INSPECTION
N
DART's line
CURB
SHOULDER
SOUTHBOUND LANE ---->
<---- NORTHBOUND LANE
M.T.O. BRIDGE 32-80
DELAMINATIONS
DISTANCE
(METERS)
PASS NUMBER
M.T.O. BRIDGE 32-80
MAP 82
SCALING
A
B
C
D
E
F
G
H
I
J
1
2
3
4
5
6
7
8
9
10
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

Delamination

Scal…

Figure 81: Delaminated and Scaled Areas at Little Bob North Bridge Detected by Penetradar.

```
$storage:2

* FL DART90 FPLATE90 REMIR90 SIGNAL90 PEAKCD90 PEAKS90 TABLE90 /link atlfor;

* DART 1990 386-system software package

************************************************************************

      PROGRAM DART90

******************* ***************************************************

      character clearscreen*4, heading*30, error*35
      character title*60, fplate*15, calib*15, iready

      parameter ( HEADING = '     [ DART SOFTWARE PACKAGE ]')
      parameter ( FPLATE  = 'FPLATE.MTO' )
      parameter ( ERROR   = ' ==> input ERROR, please try again')

      parameter ( MAXLINE = 20  )
      parameter ( NSAMPLE = 512 )
      parameter ( NTRACES = 63  )
      parameter ( Mcycle  = 126 )

      character dfile( MAXLINE )*15, datfile*15
      character condit
      integer*4 bytesize( MAXLINE )

      common /BLK1/ ref( NSAMPLE ), ydata( NSAMPLE ), zdata( NSAMPLE )
      common /BLK2/ NUMCYCLE( MAXLINE ), mmASP, mmCONC, CONDIT, ASPdc

**********************
* utilities routines *
**********************
      CLEARSCREEN = char(27)//'['//'2'//'J'          ! clear display
      write(*,*) clearscreen
      CALL GETDAT ( iyr, imon, iday )                ! date stamp
      CALL GETTIM ( ihr, imin, isec, i100th )        ! time stamp
      write(*,20)imon,iday,iyr,ihr,imin,isec,i100th
 20   format(1x,'DATE = ',i2,1x,i2,'.',i4,10x,' TIME = ',i2,3(':',i2),/)

*********
* INPUT *
*********
      write(*,'(a,/)') heading
      write(*,'(a)') ' - type BRIDGE/SITE name - '
      read(*,'(a)') title

***************
* fifth wheel *  10 cm, 1 m, 10 m
***************
      write(*,*) clearscreen
      write(*,20) imon,iday,iyr,ihr,imin,isec,i100th
      write(*,'(a,/)') heading
      write(*,'(1x,a,/)') title
 30   write(*,'(a)') '     - fifth wheel information -'
      write(*,'(a\)') ' type (0)-10 cm, (1)- 1 m, (2)-10 m : '
      read(*,*) iwheel
      if ( iwheel .lt. 0 .or. iwheel .gt. 2 ) then
          write(*,'(a,/)') error
          go to 30
      endif

*************************
* core sample (asphalt) *
*************************
      write(*,*) clearscreen
      CALL GETTIM ( ihr, imin, isec, i100th )
      write(*,20) imon,iday,iyr,ihr,imin,isec,i100th
      write(*,'(a,/)') heading
      write(*,'(1x,a,/)') title
      write(*,'(a)') '     - ASPHALT core information -'
      write(*,'(a\)') ' give thickness of asphalt core (mm) : '
      read(*,*) asphalt

********************************
* core sample (concrete cover) *
********************************
      write(*,'(1x,/,a)') '       - CONCRETE cover information -'
 40   write(*,'(a\)')     ' input (0)-automatic, (1)-operator input : '
      read(*,*) iconcrete
      if( iconcrete .ne. 0 .and. iconcrete .ne. 1 )then
         write(*,'(a,/)') error
         go to 40
      endif
      if( iconcrete .eq. 1 )then
         write(*,'(1x,/,a\)') ' give thickness of concrete cover (mm) : '
         read(*,*) concrete
      endif

****************
* no. of lines *   maximum 20 lines
****************
      write(*,*) clearscreen
      CALL GETTIM ( ihr, imin, isec, i100th )
      write(*,20) imon,iday,iyr,ihr,imin,isec,i100th
      write(*,'(a,/)') heading
      write(*,'(1x,a,/)') title
 50   write(*,'(a\)') ' number of lines (max 20) to be collected : '
      read(*,*) numline
      if( numline .le. 0 .or. numline .gt. maxline )then
         write(*,'(a,/)') error
         go to 50
      endif

***********************************
* mode: from ANALOG/DIGITAL data *
***********************************
 60   write(*,'(1x,/,a\)') ' [mode] 0 - ANALOG, 1 - data files : '
      read(*,*) mode
      if( mode .ne. 0 .and. mode .ne. 1 )then
           write(*,'(a,/)') error
           go to 60
      endif
      IF ( mode .eq. 1 ) THEN
         do I = 1, numline
              write(*,70) I
 70            format(1x,/,    ' data file (',i2,') name is : ',\)
              read(*,'(a)') dfile(i)
              write(*,'(a\)') ' number of byte          : '
              read(*,*) bytesize(i)
         enddo
      ENDIF
      write(*,'(/,a\)')' [ DEBUG ] 0-no, 1-yes : '
      read(*,*) isave
      write(*,'(/,a\)')' [ CALIBRATION ] data file name (*.MTO)      : '
      read(*,'(a)') CALIB

**************************
* flat plate information *
**************************
      write(*,*) clearscreen
      CALL GETTIM ( ihr, imin, isec, i100th )
      write(*,20) imon,iday,iyr,ihr,imin,isec,i100th
      write(*,'(a,/)') heading
      write(*,'(1x,a,/)') title
 80    write(*,'(a\)')' [ FLAT PLATE ] (0)-analog, (1)-file, (2)-input: '
      read(*,*) iplate
      if( iplate .lt.0 .and. iplate .gt. 2 )then
          write(*,'(a,/)') error
          go to 80
      else if ( iplate .eq. 2 ) then
          write(*,'(1x,/,a\)') ' flat plate amplitude (v) is : '
          read(*,*) fpamp
      else if( iplate .eq. 1 ) then
          write(*,90) fplate
 90        format(1x,/,1x,'FLAT PLATE file NAME must be : ',a15)
          write(*,'(a)')      ' -**- strike return if OKay -**-'
          write(*,'(a\)')     '         or <CTRL-C to abort>     '
          read(*,'(a)') iready
      endif

****************************************
* flat plate amplitude from ANALOG data *
****************************************
      IF( iplate .eq. 0 ) then
         write(*,*) clearscreen
         CALL GETTIM ( ihr, imin, isec, i100th )
         write(*,20) imon,iday,iyr,ihr,imin,isec,i100th
         write(*,'(a,/)') heading
         write(*,'(1x,a)') title

 100      write(*,'(1x,/,a)') '    - FLAT PLATE: check hardware -'
         write(*,'(a)')      ' -**- strike return when ready -**-'
         write(*,'(a\)')     '         or <CTRL-C to abort>     '
```

```
          read(*,'(a)') iready
          if( iready .ne. ' ' ) go to 100
        ENDIF

*=====================================================================
*| FILE = CALIB.MTO (10, OLD, UNFORMATTED, READ, FPLATE90+SIGANLAD ) |
*| open/close                                        |
*=====================================================================
* get calibration waveform
        OPEN ( 10, file = calib,  status ='old', form='unformatted' )
              rewind ( 10 )
              read ( 10 ) ( REF ( i ), i = 1, nsample )
        CLOSE ( 10 )

*************
* FPLATE90 *   digitized flat plate signals
*************
* call subroutine FPLATE90 to digitize 20 traces of metal plate signal.
      IF( iplate .ne. 2 ) CALL FPLATE90 ( IPLATE, ISAVE, FPAMP )
      write(*,*) ' '
      write(*,'(a)')     ' -**- strike return when ready -**-'
      read(*,'(a)')ready

* fifth wheel pulse
      if( iwheel .eq. 0 ) wheel =  0.10
      if( iwheel .eq. 1 ) wheel =  1.00
      if( iwheel .eq. 2 ) wheel = 10.00

*==================================================================
*|  FILE = PLOT.MTO ( 10, UNKNOWN, UNFORMATTED, WRITE, SIGNAL90 ) |
*|  open/close                                        |
*==================================================================
* create data file = PLOT.MTO for PLOT deteriorations
* store the LINE number, FIFTH wheel and no. of cycles
      OPEN ( 10, file='PLOT.MTO', status='unknown', form='formatted' )
             rewind( 10 )
             write( 10, 15 ) numline, wheel
15           format(1x,i2,1x,f5.2)

*=======================================================================
*| FILE = CONDIT.MTO ( 40, UNKNOWN, FORMATTED, WRITE, SIGANAL90       |
*| open/close                                         |
*=======================================================================
      OPEN ( 40, file='CONDIT.MTO',status='unknown',form='formatted' )
           rewind( 40 )

*=======================================================================
*| FILE = DEBUG.MTO ( 20, UNKNOWN, FORMATTED, WRITE, SIGANAL90+PEAKS90 |
*| open/close                                         |
*=======================================================================
* create data file  = DEBUG.MTO for DEBUGGING
      if ( isave .eq. 1 ) then
         OPEN( 20,file = 'DEBUG.MTO',status='unknown',form='formatted' )
           rewind( 20 )
      endif
```

```

*==========================================================================
*| FILE = TABLE.MTO ( 30, UNKNOWN, UNFORMATTED, WRITE, SIGNAL90+TABLE90 |
*| open/close                                                     |
*==========================================================================
* create data file = TABLE.MTO for SUMMARY output
      OPEN( 30, file='TABLE.MTO', status='unknown', form='unformatted' )
          rewind( 30 )

      IWAVE   = 0          ! count total waveform per site

************************
* GET A NEW LINE HERE *
************************
      DO 200 LINE = 1, NUMLINE

* get ready for radar signal
          write(*,*) clearscreen
          CALL GETTIM ( ihr, imin, isec, i100th )
          ihrS    = ihr
          iminS   = imin
          isecS   = isec
          i100thS = i100th
          write(*,20) imon,iday,iyr,ihr,imin,isec,i100th
          write(*,'(a,/)') heading
          write(*,'(1x,a,/)') title
          write(*,*) 'flat plate amplitude = ',fpamp,' volt'
          write(*,*) ' '
          write(*,*) ' -*-*- get LINE number : ',LINE,' -*-*-'

        IF ( mode .eq. 0 ) THEN

110        write(*,'(1x,/,a)')' - RADAR SIGNAL: check hardware -'
          write(*,'(a)')     ' -**- strike return when ready -**-'
          write(*,'(a\)')    '        or <CTRL-C to abort>    '
          read(*,'(a)') iready
          if( iready .ne. ' ' ) go to 110

          if( wheel .eq. 0.10 )then
             write(*,'(1x,/,a,/)') '     sample at every 10 cm '
          else
             write(*,'(1x,/)') '    sample at every',wheel,' m'
          endif

        ELSE

          datfile          = dfile ( Line )
          NUMCYCLE( Line ) = int2 (bytesize(Line)/int4(1024))

        ENDIF

*************
* SIGNAL90 *  subroutine SIGNAL90 to digitized/real signal and process
*************
* call subroutine SIGNAL90 to digitize radar data
```

```
                CALL SIGNAL90 ( title, mode, line, datfile, isave, fpamp,
     +                        asphalt, iconcrete, concrete, iwave, wheel )

                Length = int( float( numcycle( Line ) ) * wheel )

                write ( 10, 16 ) Line, numcycle(Line)
      16         format(1x,i2,1x,i5)

                open( 90, file = 'PRN')
                write(*, 120) Line, NUMCYCLE( Line ), LENGTH
                write(90,120) Line, NUMCYCLE( Line ), LENGTH
      120        format(1x,/,' Line = ',i2,4x,'total TRACE = ',i5,4x,
     +                'bridge LENGTH = ',i5,' metres',/)

                write(*, '(a\)') ' START  time: '
                write(90,'(a\)') ' START  time: '
                write(*, 124) imon,iday,iyr,ihrS,iminS,isecS,i100thS
                write(90,124) imon,iday,iyr,ihrS,iminS,isecS,i100thS
      124        format(1x,'DATE = ',i2,1x,i2,'.',i4,10x,'TIME = ',i2,3(':',i2))
                CALL GETTIM ( ihr, imin, isec, i100th )
                write(*, '(a\)') ' FINISH time: '
                write(90,'(a\)') ' FINISH time: '
                write(*, 124) imon,iday,iyr,ihr,imin,isec,i100th
                write(90,124) imon,iday,iyr,ihr,imin,isec,i100th
                close( 90 )
                write(*,*)' '
                write(*,'(a\)') ' -**- strike return when ready -**-'
                read(*,'(a)') ready

      200    CONTINUE

            close ( 10 )
            if ( isave .eq. 1 ) close ( 20 )
            close ( 30 )
            close ( 40 )

***********
* TABLE90 *  print results in table form
***********
* prepare to print TABLE
            CALL TABLE90 (TITLE, NUMLINE, FPAMP, ASPHALT, ICONCRETE,
     +                    CONCRETE, WHEEL)

            write(*,*) ' '
            write(*,'(a\)') ' -**- strike return when ready -**-'
            read(*,'(a)') ready
            write(*,*) clearscreen
            CALL GETTIM ( ihr, imin, isec, i100th )
            write(*,20) imon,iday,iyr,ihr,imin,isec,i100th
            write(*,'(a,/)') heading
            write(*,'(1x,a,/)') title
            write(*,'(1x,a,/)') ' [ data files information ]'
            write(*,'(1x,a\)') ' SAVE the following DATA FILEs BEFORE the'
            write(*,'(1x,a,/)')' NEXT run'
            write(*,'(1x,a)')' PLOT.MTO      (formatted)   : Line infomation'
      write(*,'(1x,a)')' CONDIT.MTO    (formatted)   : bridge condition'
      write(*,'(1x,a)')' SUMMARY.MTO   (formatted)   : summary'
      write(*,'(1x,a)')' TABLE.MTO     (unformatted) : BKUP summary'

***************
* data files *
***************
      if ( mode .eq. 0 )write(*,'(1x,a)')
     +                   ' drive Es files (binary)     : signal raw'

      IF ( ISAVE .EQ. 1 ) THEN
       write(*,'(1x,a)')' SIGSYNC.MTO   (unformatted) : signal sync'
       write(*,'(1x,a)')' SIGRESDUE.MTO (unformatted) : signal resdue'
       write(*,'(1x,a)')' DEBUG.MTO     (formatted)   : debug'
       if ( iplate .eq. 0 )write(*,'(1x,a)')
     +                   ' FPLATE.MTO    (binary)      : fplate raw'
       if ( iplate .ne. 2 ) then
        write(*,'(1x,a)') ' FPSYNC.MTO    (unformatted) : fplate sync'
        write(*,'(1x,a)') ' FPRESDUE.MTO  (unformatted) : fplate resdue'
        write(*,'(1x,a)') ' PLATE.MTO     (unformatted) : fplate signal'
       endif
      ENDIF

      END
```

```
$storage:2

* fl/c FPLATE90.for

* obtain the flat plate reflection

      SUBROUTINE FPLATE90 ( iplate, isave, fpamp )

$INCLUDE:'ATLDEFS.FOR'
$INCLUDE:'ATLERRS.FOR'

      PARAMETER ( NBUFFERS = 1      )  ! no. of data buffers
      PARAMETER ( NBUFSIZE = 512    )  ! data buffer size (WORDS)
      PARAMETER ( NXMBUF   = 1      )  ! no. of XM buffers
      PARAMETER ( NXMSIZE  = 10240 )  ! XM buffer size (words)
      PARAMETER ( NTRACES  = 63     )
      PARAMETER ( N325     = 325    )

      INTEGER*2 STATUS
      INTEGER*2 ADCHANNELS(16), ADGAINS(16)
      INTEGER*2 SCANCOUNT, UNITNUMBER, RELEASEDBUFFER
      INTEGER*2 BUFS(NBUFSIZE,NBUFFERS), BUFNUM(NBUFFERS,2)
      INTEGER*2 XMBUFNUM(NXMBUF)
      INTEGER*2 BLENGTH, TIMEOUT
      INTEGER*2 SAMPLE

      REAL*4 RATE
      character key, plate*15

      common /BLK1/ ref( NBUFSIZE ), ydata ( NBUFSIZE ), zdata(NBUFSIZE)
      dimension xdata( NBUFSIZE)
      DATA PLATE/ 'E:FPLATE.MTO' /

*  BEGIN PROGRAM
      WRITE(*,'(1x,/,a)') '       - FLAT PLATE -'

****************
* ANALOG DATA * call ATLAB's routines to sample flat plate signal
****************
* for iplate = 0 (from tape)
      IF ( IPLATE .EQ. 0 ) THEN

* set hardware parameters
      unitnumber   = 1
      ItrigV       = 100        ! trigger threshold
      ItrigC       = 1          ! A/D channel # 1 for tigger
      RATE         = 25000.     ! 25 KHz sampling rate

*  INITIALIZE THE ATLAB SUBROUTINES, select board, reset device
      write(*,'(a)')'  - initialize DT2821 and ATLAB routines -'
      STATUS = ALINIT ()
      STATUS = ALSB (UNITNUMBER)
      STATUS = ALRSET ()

* REQUEST SCAN COUNT, CHANNELS, GAINS AND COMMENTS.
```

```
      SCANCOUNT      = 1
      ADCHANNELS(1) = 0   ! A/D channel # 0 for radar signal
      ADGAINS(1)     = 1

* ASSIGN USER BUFFER and link it to buffer transfer list
      STATUS = ALFDL ( BUFS(1,1), BLENGTH )
      STATUS = ALDB( BUFNUM(1,2), BUFS(1,1), NBUFSIZE )
      STATUS = ALLB( BUFNUM(1,2) )

* set XM buffer, note that the size is NXMSIZE
      STATUS = ALAXMB ( XMBUFNUM(1), NXMSIZE )

*  SET THE TIMEOUT RATE BASED ON THE SAMPLING RATE AND THE BUFFER SIZE
      TIMEOUT = ( NINT( REAL(NBUFSIZE)/RATE ) + 1 ) * 2 * NBUFFERS
      STATUS  = ALST ( TIMEOUT )

      write(*,10) NXMSIZE
10     format(1x,' - EXPECTED to acquire maximum ',i10,' SAMPLES  -')

*   BEGIN BURST OPERATION
      write(*,*)' - ACQUIRING DATA [wait] -'

* skip the first 20 triggers
      I = 0
20     I = I + 1

* Trigger is HIGH wait for it to go low
30     STATUS = ALAV ( ItrigC, 1, sample )
      IF ( sample .GT. ItrigV .and. sample .lt. 2048 ) GO TO 30

* Trigger is LOW wait for it to go HIGH
40     STATUS = ALAV ( ItrigC, 1, sample )
      IF ( sample .LE. ItrigV .or. sample .ge. 2048  ) GO TO 40

      IF ( I .LT. 20 ) GO TO 20

* set up parameters for data acquisition
      IBUF      = bufnum(1,2)        ! user buffer number
      ICYCLE    = NXMSIZE/NBUFSIZE   ! cycle counter (512s/c)

* start each cycle of 512 samples here
      icount = 0                     ! counter for 32K words
      I      = 0
50     I      = I + 1

**************
** TRIGGER **   channel 1
**************
* Trigger is HIGH wait for it to go low
60     STATUS = ALAV ( ItrigC, 1, sample )
      IF ( sample .GT. ItrigV .and. sample .lt. 2048 ) GO TO 60

* Trigger is LOW wait for it to go HIGH
70     STATUS = ALAV ( ItrigC, 1, sample )
      IF ( sample .LE. ItrigV .or. sample .ge. 2048  ) GO TO 70
```

```

******************
** RADAR pulse **      channel 0
******************
* set DMA parameters
      STATUS = ALSETA (INTTRIGGER+INTCLOCK,SCANCOUNT,ADCHANNELS,ADGAINS)
      STATUS = ALSF ( RATE )
      STATUS = ALBAD ()

* wait for the user buffer to complete DMA
      STATUS = ALWFC ( ibuf )
      STATUS = ALSTOP ()

* release the buffer from DMA
      STATUS = ALRELB ( UNITNUMBER, RELEASEDBUFFER )

* copy user buffer to XM buffer
      STATUS = ALCB ( RELEASEDBUFFER, XMBUFNUM(1), 0, icount, NBUFSIZE )

* return buffer to DMA
      STATUS = ALRETB ( releasedbuffer )

* updata counters
      icount = icount + nbufsize                    ! 32256 yet

* if cycle NE 63, get more
        if ( I .LT. ICYCLE ) go to 50
        write(*,*) ' - no. of waveforms have been acquired = 20 -'

      ELSE

**************
* DATA FILE *     FPLATE.MTO 20 cycles
**************
* for iplate = 1 (read data file FPALTE.MTO)
        ICYCLE = 20

      ENDIF

      write(*,'(1x,a)')' - processing data [wait] -'

* initialize vectors
      do i = 1, nbufsize
            ydata(i) = 0.0
            xdata(i) = 0.0
      enddo

*---------------------------------------------------------------------
*| FILE = PLATE = E:FPLATE.MTO ( 40, UNKNOWN, BINARY, READ ) |
*| open/close                                                |
*---------------------------------------------------------------------
* access data file
      OPEN ( 40, file = PLATE, status='unknown', form='binary' )
            REWIND ( 40 )

```

```
*=================================================================
*! FILE = FPSYNC.MTO ( 50, UNKNOWN, UNFORMATTED, WRITE )    !
*! open/close                                 !
*=================================================================
* access data file
      if ( isave .eq. 1 ) then
      OPEN(50, file ='FPSYNC.MTO', status='unknown', form='unformatted')
          REWIND ( 50 )
*=================================================================
*! FILE = FPRESDUE.MTO ( 60, UNKNOWN, UNFORMATTED, WRITE ) !
*! open/close                                 !
*=================================================================
      OPEN(60,file ='FPRESDUE.MTO',status='unknown',form='unformatted')
          REWIND ( 60 )
      endif

* get a new cycle of 512 samples
      icount = 0
      iskip  = 0
      DO 200 I = 1, ICYCLE

* copy XM buffer to user buffer
          IF ( iplate.eq.0 ) THEN
            STATUS = ALCB ( XMBUFNUM(1),BUFNUM(1,2),icount,0,NBUFSIZE )

* SAVE flat plate RAW DATA opened in FPLATE90.for
            WRITE ( 40 ) (BUFs( j, 1 ), j = 1, nbufsize )

          ELSE

*******************
* READ DATA FILE *
*******************
* READ flat plate data file opened in FPLATE90.for
            READ ( 40 ) ( BUFs( j, 1 ), j = 1, nbufsize )

          ENDIF

* copy user buffer to vector
          sum    = 0.0
          valmin = 0.0
          Nsync  = 55
          iflag  = 0

          do j = 1, nbufsize
               zdata(j) = float( bufs (j,1) ) * 10./2048.
               sum      = sum + zdata(j)

* locate the first zero crossing sample = nsync
               if( zdata(j) .lt. valmin .and. iflag .eq. 0 )then
                    iflag = 1
                    Nsync = j
               endif
          enddo

```

```
* locate the reference point (the first valley) which is around 40 samples
* from first zero crossing
          if( iflag .eq. 1 )then
              do j = Nsync, Nsync + 30
                    if( zdata(j) .le. valmin ) then
                        valmin = zdata(j)
                        Msync  = j
                    endif
              enddo
          else
              Msync = 55
          endif

* zero average each waveform
* and subtract the internal reflection
          sum = sum/float(nbufsize)
          nref = nbufsize - Msync + 1
          do j = 1, nref
                  ydata(j) = zdata(Msync+j-1)-sum
          enddo

* SAVE SYNC data in FPSYNC.MTO opened in FPLATE90.for
          if ( isave .eq. 1 ) write ( 50 ) ( ydata (j), j=1,nbufsize )

* remove internal reflection
          CALL REMIR90 ( maxpka, pkamax )

* SAVE RESDUE data in FPRESDUE.MTO opened in FPLATE90.for
          if ( isave .eq. 1 ) write ( 60 ) ( ydata (j), j=1,nbufsize )

* throw away the bad sampled waveform
          if ( pkamax .le. 2.5 .or.
     +            maxpka .le. 125 .or.
     +            maxpka. ge. 150 ) go to 100

* line up all waveforms wrt to index 135 (5.4 ns)
          mref = 135
          il   = 0
          do j = maxpka, n325
                il = il + 1
                k  = mref + il - 1
                if ( k .le. n325 ) zdata(k) = ydata(j)
          enddo

          il = 0
          do j = mref-1, 1, -1
                il = il + 1
                k  = maxpka - il
                if ( k .gt. 0 ) zdata(j) = ydata(k)
          enddo

          do j = 1, n325
                xdata(j) = xdata(j) + zdata(j)
          enddo

```

```
                go to 110
      100        iskip = iskip + 1
                write(*,*)' - waveform # ',i,' is corrupted -'

      110        icount = icount + nbufsize

      200   CONTINUE

            CLOSE ( 40 )
            if ( isave .eq. 1 ) then
                CLOSE ( 50 )
                CLOSE ( 60 )
            endif

            if( iskip.eq.icycle )then
              write(*,'(1x,/,a\)') ' - data are corrupted, abort program - '
              read(*,'(a)')key
            endif

      * average over ICYCLE waveforms and get amplitude
            fpamp = 0
            do i = 1, nbufsize
                  ydata(i) = xdata(i)/float(icycle-iskip)
                  if( ydata(i) .ge. fpamp ) fpamp = ydata(i)
            enddo
            write(*,*)' flat plate amplitude = ', FPAMP

      *==========================================================
      *| FILE = PLATE.MTO ( 50, UNKNOWN, UNFORMATTED, WRITE ) |
      *| open/close                                |
      *==========================================================
      * and this is for MATLAB plot
            if ( isave .eq. 1 ) then
            open( 50, file='PLATE.MTO',status='unknown',form='unformatted' )
                rewind ( 50 )
                write  ( 50 ) ( ydata(i), i = 1, nbufsize )
            close ( 50 )
            endif

      *  TERMINATE ATLAB OPERATIONS
            write(*,*)' - FLAT PLATE DONE -'

            if( iplate .eq. 0 )then
               STATUS = ALSTOP ()
               STATUS = ALTERM ()
            endif

            RETURN
            END
```

```

$storage:2

* fl/c REMIR90.for

      SUBROUTINE REMIR90 ( MAXPKA, PKAMAX )

      common /BLK1/ ref(512), ydata(512), zdata(512)

* initialization
      resex  = 0.0
      reslft = 0.0
      resrgt = 0.0
      nopt   = 116
      n325   = 325

* measure the residue magnitude of the 116 samples
      DO  J = 1,nopt

* subtraction at exact line up
              exact = ydata(j) - ref(j)
              resex = resex + (exact*exact)

* subtraction at shift 1 sample to the left
              rleft  = ydata(j+1) - ref(j)
              reslft = reslft + (rleft*rleft)

* subtraction at shift 1 sample to the right
              right  = ydata(j) - ref(j+1)
              resrgt = resrgt + (right*right)

      ENDDO

* calculate RMS values
      resex  = (resex/float(nopt))**0.5
      reslft = (reslft/float(nopt))**0.5
      resrgt = (resrgt/float(nopt))**0.5

* compare the three residues and select the minimum
      resmin = min( resex, reslft, resrgt )

* calculate the residue waveform
      DO J = 1, n325

* if resmin=resex, calculate the residue waveform
            if ( resmin .eq. resex )  zdata(j) = ydata(j) - ref(j)

* if resmin=reslft, find the residue signal
            if ( resmin .eq. reslft ) zdata(j) = ydata(j+1) - ref(j)

* if resmin=resrgt, find the residue signal
            if ( resmin .eq. resrgt ) zdata(j) = ydata(j) - ref(j+1)

      ENDDO

```

```
* adjust the residue signal and search for the surface echo peak
      pkamax = 0
      maxpka = 135              ! default value
      DO J = 1,n325
           ydata(j) = zdata(j)
           if ( ydata(j) .ge. pkamax ) then
               PKAMAX = ydata(j)
               MAXPKA = j
           endif
      ENDDO

      RETURN
      END
```

```
$STORAGE:2

* FL/c SIGNAL90.for

* DART's data acquisition

**********************************************************************
      SUBROUTINE SIGNAL90 ( TITLE, MODE, LINE, DATFILE, ISAVE, FPAMP,
     +                  ASPHALT, ICONCRETE, CONCRETE, IWAVE, WHEEL )
**********************************************************************
$INCLUDE:'ATLDEFS.FOR'
$INCLUDE:'ATLERRS.FOR'

        PARAMETER (NBUFFERS  = 1     ) ! no. of data buffers
        PARAMETER (NBUFSIZE  = 512   ) ! data buffer size (WORDS)
        PARAMETER (Nopt      = 325   ) ! waveform sample
        PARAMETER (NXMBUF    = 119   ) ! no. of XM buffers
        PARAMETER (NXMSIZE   = 32256 ) ! XM buffer size (words)
        PARAMETER (MCYCLE    = 126   ) ! data buffer size (waveforms)
        PARAMETER (MAXLINE   = 20    ) ! max 12 passes
        PARAMETER (PTAVE     = 5     ) ! 5-point averager

        INTEGER*2 ADCHANNELS(16)      ! channel scan list
        INTEGER*2 ADGAINS(16)         ! gains for channels

*  LOCAL VARIABLES
       INTEGER*2 STATUS
       INTEGER*2 SCANCOUNT, UNITNUMBER, RELEASEDBUFFER
       INTEGER*2 TIMEOUT
       INTEGER*2 BUFS(NBUFSIZE,NBUFFERS), BUFNUM(NBUFFERS,2)
       INTEGER*2 BLENGTH
       INTEGER*2 XMBUFNUM(NXMBUF)
       INTEGER*2 SAMPLE
       INTEGER*4 IFULL, I64Kw, inewXM, IXMBUF(NXMBUF)
       INTEGER*4 isumASP, isumCONC

       CHARACTER TITLE*60
       CHARACTER RFILE(MAXLINE)*15, DATFILE*15, dfile*15
       CHARACTER CONDIT

       REAL*4 RATE

       common /BLK1/ ref(NBUFSIZE), ydata(NBUFSIZE), zdata(NBUFSIZE)
       common /BLK2/ NUMCYCLE( MAXLINE ), mmASP, mmCONC, CONDIT, ASPdc

       dimension wdata( Nbufsize, 10 ), mxC( 10 ), mxD( 10 )

* store RAW data on DRIVE E:
      DATA RFILE(1) / 'E:LINE1.MTO'/,  RFILE(2)/'E:LINE2.MTO'/
      DATA RFILE(3) / 'E:LINE3.MTO'/,  RFILE(4)/'E:LINE4.MTO'/
      DATA RFILE(5) / 'E:LINE5.MTO'/,  RFILE(6)/'E:LINE6.MTO'/
      DATA RFILE(7) / 'E:LINE7.MTO'/,  RFILE(8)/'E:LINE8.MTO'/
      DATA RFILE(9) / 'E:LINE9.MTO'/,  RFILE(10)/'E:LINE10.MTO'/
      DATA RFILE(11)/'E:LINE11.MTO'/,  RFILE(12)/'E:LINE12.MTO'/
      DATA RFILE(13)/'E:LINE13.MTO'/,  RFILE(14)/'E:LINE14.MTO'/
      DATA RFILE(15)/'E:LINE15.MTO'/,  RFILE(16)/'E:LINE16.MTO'/
      DATA RFILE(17)/'E:LINE17.MTO'/,  RFILE(18)/'E:LINE18.MTO'/
      DATA RFILE(19)/'E:LINE19.MTO'/,  RFILE(20)/'E:LINE20.MTO'/

* get data from data files
      IF( MODE .EQ. 1 ) GO TO 200

****************
* ANALOG DATA *  use ATLAB's routines
****************
* otherwise get data from analog data
      UNITNUMBER = 1
      IsyncV     = 100       ! line sync threshold
      I5thwV     = 100       ! fifth wheel threshold
      ItrigV     = 100       ! trigger threshold

      IsyncC     = 3         ! A/D channel # 3 for sync
      I5thwC     = 2         ! A/D channel # 2 for 5th wheel
      ItrigC     = 1         ! A/D channel # 1 for tigger

      RATE       = 25000.    ! sampling rate

*  BEGIN PROGRAM
      write(*,*) '          - DART data acquisition - '
      write(*,*) '  can digitize 15Mb (max) data using DMA burst mode'
      write(*,*) '        each radar waveform has 512 samples'
      write(*,*) ' '

*  INITIALIZE THE ATLAB SUBROUTINES, select board, reset device
      write(*,*)' - initialize DT2821 and ATLAB routines [wait] - '
      STATUS = ALINIT ()
      STATUS = ALSB (UNITNUMBER)
      STATUS = ALRSET ()
      STATUS = ALSF ( RATE )

* REQUEST SCAN COUNT, CHANNELS, GAINS AND COMMENTS.
      SCANCOUNT = 1
      ADCHANNELS(1)=0             ! A/D channel # 0 for radar signal
      ADGAINS(1) = 1

* DECLARE USER BUFFER
      STATUS = ALFDL ( BUFS(1,1), BLENGTH )
      STATUS = ALDB ( BUFNUM(1,2), BUFS(1,1), NBUFSIZE )
      STATUS = ALLB ( BUFNUM(1,2) )

* set XM buffer, note that the size is NXMSIZE*2
      do i=1,nxmbuf
         STATUS = ALAXMB ( XMBUFNUM(i), NXMSIZE*2 )
      enddo

*  SET THE TIMEOUT RATE BASED ON THE SAMPLING RATE AND THE BUFFER SIZE
      TIMEOUT =(NINT(REAL(NBUFSIZE)/RATE)+1)*2*NBUFFERS
      STATUS = ALST ( TIMEOUT )

***********
```

```
** SYNC **  channel 3
**********
* wait for line SYNC channel 3
        write(*,*)' - line SYNC is NOT READY  [wait] -'
50       STATUS = ALAV ( IsyncC, 1, sample )
        if(sample .lt. IsyncV .or. sample .ge. 2048 ) GO TO 50
        write(*,*)' - line SYNC is now READY -'

        i64Kw=int4(nxmsize)*2      ! each XM-buffer = 64512 words
        iFull=int4(nxmbuf-1)*i64Kw ! fill up 118 XM-buffers = 7.6Mw
        write(*,60)iFULL
60       format(1x,' - EXPECTED to acquire maximum ',i10,' SAMPLES  -')

* BEGIN BURST OPERATION
        write(*,*)' - ACQUIRING DATA [wait] -'

        IBUF  = bufnum(1,2)              ! user buffer number
        LIMIT = NXMSIZE                    ! 32256 samples
        ICYCLE= (NXMSIZE/NBUFSIZE)*2    ! cycle counter (512s/c)

* START a NEW XM-BUFFER
        IXM    = 2        ! initial XM-buffer

70       CONTINUE
        inewXM =0         ! inewXM = XM-buffer counter
        icount =0       ! icount = counter for 32K words
        iflag  =0       ! iflag  = flag for over 32K words

* START each CYCLE of 512 samples here
        I = 1
80       CONTINUE

****************
** 5th WHEEL **    channel 2
****************
* FIFTH wheel pulse is HIGH wait for it to go LOW
90           STATUS = ALAV ( I5thwC, 1, sample )
            if( sample .GT. I5thwV .and. sample .lt. 2048 ) GO TO 90

* check for line SYNC is still HIGH otherwise stop
100          STATUS = ALAV ( IsyncC, 1, sample )
            if( sample .lt. isyncV .or. sample .ge. 2048 )go to 150

* FIFTH wheel pulse is LOW wait for it to go HIGH
            STATUS= ALAV ( I5thwC, 1, sample )
            if( sample .LE. I5thwV .or. sample .ge. 2048 ) GO TO 100

**************
** TRIGGER **   channel 1
**************
* Trigger is HIGH wait for it to go low
110          STATUS = ALAV ( ItrigC, 1, sample )
            if( sample .GT. ItrigV .and. sample .lt. 2048 ) GO TO 110

* Trigger is LOW wait for it to go HIGH
```

```
120        STATUS= ALAV ( ItrigC, 1, sample )
          if( sample .LE. ItrigV .or. sample .ge. 2048 ) GO TO 120

******************
** RADAR pulse **      channel 0
******************
* set DMA parameters
        STATUS=ALSETA( INTTRIGGER+INTCLOCK,SCANCOUNT,ADCHANNELS,ADGAINS)
        STATUS = ALBAD ()

* wait for the user buffer to complete DMA
        STATUS = ALWFC ( ibuf )
        STATUS = ALSTOP()

* release the buffer from DMA
        STATUS=ALRELB(UNITNUMBER,RELEASEDBUFFER)

* copy user buffer to XM buffer
        STATUS=ALCB( RELEASEDBUFFER, XMBUFNUM(ixm), 0, icount, NBUFSIZE)

* return buffer to DMA
        STATUS=ALRETB(releasedbuffer)

* updata counters
        icount       = icount+nbufsize            ! 32256 yet
        inewXM       = inewXM+int4(nbufsize)      ! 64512 yet
        ixmbuf(ixm) = inewXM
        ilastxm      = ixm

* if top 32K XM-buffer filled up then
* transfer these 32K samples to the lower XM buffer
        if((icount.ge.limit).and.(iflag.eq.0))then
          STATUS = ALCB(XMBUFNUM(ixm),XMBUFNUM(ixm),0,icount,icount)
          icount = 0
          iflag  =1
        endif

**********
** SYNC **   channel 3
**********
* detect line sync low
        STATUS = ALAV ( IsyncC, 1, sample )
150      if( sample .lt. IsyncV. or. sample .ge. 2048 )then
          ilastxm=ixm
          ixm=nxmbuf       ! terminate XM-buffer counter
          i=icycle         ! terminate cycle counter
          write(*,*)'  '
          write(*,*)'  - line SYNC is LOW -'
          write(*,*)' - STOP acquiring data -'
        endif

* if cycle NE 126, get more
        I= I + 1
        IF ( I .LE. ICYCLE ) GO TO 80
```

```
* fill up all XM yet, if not get more
        IXM= IXM + 1
        IF ( IXM .LE. NXMBUF ) GO TO 70

* calculate the total no. of sample being acquired
        iFULL=0
        do i=2,ilastxm
            iFULL = iFULL + ixmbuf(i)
        enddo

        DFILE = RFILE(line)          ! save raw data on drive E's files
        NUMCYCLE( Line ) = int(float(iFULL)/512.)
        IXM              = 2         ! current XM-buffer

*****************
200    CONTINUE     !  get data from file
*****************
* get parameters
        IF ( mode .EQ. 1 ) THEN
            DFILE   = DATFILE
            icycle  = MCYCLE              ! 126 waveforms each processing
            ncycle  = numcycle( Line )/icycle   ! calculate num of 126
            reminder= mod( float(numcycle(Line)), float(icycle) )
            Lcycle  = int( reminder )
            ixm     = 0
        ENDIF

        write(*,*)' '
        write(*,*)' - # of CYCLEs digitized = ',numcycle( Line ),' - '

        if( isave .eq. 1 .or. mode .eq. 1 ) then
           write(*,*)' '
           write(*,*)' - accessing data file ',DFILE,' [wait] -'
        endif
        write(*,*)' '

*=========================================================================
*| FILE = DFILE = E:Line#.MTO ( 60, UNKNOWN, BINARY, READ/WRITE ) |
*| open/close                                                     |
*=========================================================================
        OPEN (60, file=DFILE, status = 'unknown', form = 'binary')
            rewind ( 60 )

*=================================================================
*| FILE = SIGSYNC.MTO ( 70, UNKNOWN, UNFORMATTED, WRITE ) |
*| open/close                                             |
*=================================================================
        IF ( isave .eq .1 ) THEN
         OPEN(70,file='SIGSYNC.MTO',status='unknown',form='unformatted')
         rewind ( 70 )

*================================================================
*| FILE = SIGRESDUE.MTO ( 80, UNKNOWN, UNFORMATTED, WRITE ) |
*| open/close                                               |
*================================================================
        OPEN(80,file='SIGRESDUE.MTO',status='unknown',
     +          form='unformatted')
        rewind ( 80 )
      ENDIF

****************************************
* initialize parameters for processing *
****************************************
* parameters for EACH line
      Nwave   = 0          ! number of waveform per line
      Mwave   = 0
      XLoc    = 0.0        ! fifth wheel location
      LargeDC = 0          ! clear large DC counter
      SUMdc   = 0.0        ! clear dc values
      write(*,*) ' '

* parameters for AVERAGED THICKness and CONDITions
      isumASP  = 0         ! asphalt parameters
      minASP   = 500
      maxASP   = 0

      isumCONC = 0         ! concrete cover parameters
      minCONC  = 500
      maxCONC  = 0

      iGOOD    = 0         ! parameters for conditions
      iSCALED  = 0
      iDEBOND  = 0
      iDELAMIN = 0
      iCORRUPT = 0

* for 10 cm average over 1 m distance
      itenCM   = 10
      i10CM    = 0
      moneASP  = 0
      moneCONC = 0
      iBAD     = 0

* for 10 waveforms averaging
      i10WAVE  = 0
      iaveC    = 0
      iaveD    = 0
      iLL      = 0

*************************************
* get NEW XM-buffer for ANALOG DATA *
*************************************
130   CONTINUE

      IF ( mode .eq. 0 ) THEN

* ANALOG data
* calculate individual XM-buffer length and left over
          if(ixmbuf(ixm).ge.int4(LIMIT))then
            ibegin=LIMIT          ! first sample
```

```
                 ichunk=LIMIT          ! number of samples each time
              else
                 ibegin=0
                 ichunk=int2(ixmbuf(ixm))
              endif

* transfer lower XM(ixm) to upper XM(1)
              STATUS= ALCB ( XMBUFNUM(ixm),XMBUFNUM(1),ibegin,0,ichunk )

* get a new cycle of 512 samples
              icount = 0
              iflag  = 0
              left   = 0
              icycle = int(float(ixmbuf(ixm))/512.)

          ELSE

* for DATA FILE
              if ( ixm .eq. ncycle ) icycle = Lcycle   ! the reminding bits
              if ( icycle .le. 0 ) go to 400
          ENDIF

******************
* process signal *
******************
          DO 500 I = 1, ICYCLE

               Itrace = i              ! current cycle counter
               Iwave  = Iwave + 1      ! accumulated waveforms counter
               Nwave  = Nwave + 1      ! current line waveforms counter

               write(*,300) Line, Iwave, Nwave, itrace
300             format(1H+,1x,' Line = ',i2,2x,' total = ',i5,2x,
     +                     ' current = ',i5,2x,' cycle = ',i5)

****************
* get the data *
****************
               IF ( mode .EQ. 0 ) THEN

* for ANALOG data
                  STATUS=ALCB(XMBUFNUM(1),BUFNUM(1,2),icount,0,NBUFSIZE)

* SAVE the RAW data on DRIVE E:'s,  data files opened in SIGNAL90.for
                  WRITE (60)(bufs(j,1),j=1,nbufsize)

               ELSE

* read data from files opened in SIGNAL90.for
                  READ ( 60 ) (bufs(j,1),j=1,nbufsize)

               ENDIF

******************************
* remove internal reflection *
```

```
******************************
* sync the reference point for removing internal reflection
          sum    = 0.0
          Nsync  = 55
          valmin = 0.0
          izero  = 0

          do j = 1, nbufsize
             ydata(j) = 0.0
             zdata(j) = float(bufs(j,1))*10./2048.
             sum      = sum + zdata(j)
             if ( zdata(j) .lt. valmin .and. izero .eq. 0 ) then
                izero  = 1
                Nsync  = j
             endif
          enddo

          if ( izero .eq. 1 ) then
             do j = Nsync, Nsync + 30
                  if( zdata(j) .le. valmin ) then
                     valmin = zdata (j)
                     Msync = j
                  endif
             enddo
          else
             Msync = 55
          endif

          sum  = sum/float(nbufsize)
          nref = nbufsize - Msync + 1
          do j = 1, nref
             ydata(j) = zdata(Msync+j-1) - sum
          enddo

* SAVE the sync data in SIGSYNC.MTO opened in SIGNAL90.for
         if ( isave .eq. 1) write(70) (ydata(j),j=1,nbufsize)

* remove internal reflections
         CALL REMIR90 ( maxpka, pkamax )

* save residue data (SIGRESDUE.MTO) opened in signal90.for
         if ( isave .eq. 1) write( 80 )( ydata(j), j = 1, nbufsize )

****************************************
* parameters for 10 waveforms averaging *
****************************************
         i10WAVE  = i10WAVE + 1
         iTENWave = 10
         Nave     = ( numcycle(Line)/iTENWave ) * iTENWave
         if ( Nwave .gt. Nave ) iTENWave = ( numcycle(line) - Nave )
         if ( wheel .ne. 0.10 ) iTENWave = 1       ! for fifth wheel 1m/10m

         do j = 1, nbufsize
              wdata( j, i10Wave ) = ydata(j)
         enddo
```

```

      CALL PEAKCD90( fpamp, maxpks, asphalt, iconcrete, concrete,
     +               LocC, LocD )

      mxC(i10Wave) = LocC
      mxD(i10Wave) = LocD

      if ( LocC .ne. 999 ) then
           iaveC = iaveC + LocC
           iaveD = iaveD + LocD
      else
           iLL   = iLL   + 1
      endif

      IF ( i10Wave .eq. iTENWave ) THEN

          if ( i10Wave .ne. iLL ) then
               iaveC = iaveC/(i10Wave - iLL)
               iaveD = iaveD/(i10Wave - iLL)
          else
               iaveC = 999
               iaveD = 999
          endif

          LocC1 = 0
          LocD1 = 0
          do J = 1, i10Wave
                if ( iaveC .ne. 999 .and. mxC(J) .ne. 999 )then
                     idiff = ABS ( iaveC - mxC(J) )
                     LocC1 = LocC1 + idiff
                     idiff = ABS ( iaveD - mxD(J) )
                     LocD1 = LocD1 + idiff
                endif
          enddo

          if ( i10Wave .ne. iLL ) then
               LocC1 = LocC1/(i10Wave - iLL)
               if ( LocC1 .eq. 0 ) LocC1 = 5
               LocC2 = LocC1

               LocD1 = LocD1/(i10Wave - iLL)
               if ( LocD1 .eq. 0 ) LocD1 = 5
               LocD2 = LocD1
        endif

        DO 550 J = 1, i10Wave

               do k = 1, nbufsize
                      ydata (k) = wdata( k,j )
               enddo

               i10count = J
               mxC1      = mxC(j)
               mxD1      = mxD(j)

```

```
                         CALL PEAKS90 ( fpamp, iwave, itrace, i10Wave,
     +                                  i10count, mxC1, mxD1, iaveC,
     +                                  iaveD, LocC1, LocC2, LocD1,
     +                                  LocD2, permasp, isave )

**************************
* for 10 cm fifth wheel *
**************************
* average over 1m distance
          IF ( WHEEL .eq. 0.10 ) THEN

             i10CM = i10CM + 1  ! count 10 cm averaging

             if ( i10CM .le. itenCM .and. mmASP .ne. 999 ) then
                moneasp  = moneasp  + mmASP
                moneconc = moneconc + mmCONC
             else
                ibad = ibad + 1
             endif

             if ( i10CM .eq. itenCM ) then
                if ( ibad .ne. i10CM ) then
                   maveASP  = moneasp /float( i10CM-ibad )
                   maveCONC = moneconc/float( i10CM-ibad )
                else
                   maveASP  = 999
                   maveCONC = 999
                endif

                i10CM    = 0           ! reset counter after i10 = 10
                ibad     = 0
                moneASP  = 0
                moneCONC = 0

* save 1m average thickness ( TABLE.MTO ) opened in MTO90.for
                write ( 30 ) maveASP, maveCONC
             endif

          ELSE

***********************************
* for 1 m or 10 m fifth wheels *
***********************************
* save 1 and 10 m fifth wheel  ( TABLE.MTO ) opened in MTO90.for
                write( 30 ) mmASP, mmCONC

          ENDIF

* calculate sum/min/max ASPHALT thickness of current line
             IF ( mmASP .ne. 999 ) THEN
                isumASP = isumASP + int4( mmASP )
                if( mmASP .le. minASP ) minASP = mmASP
                if( mmASP .ge. maxASP ) maxASP = mmASP

* calculate sum/min/max CONCRETE cover of current line
```

```
                  isumCONC = isumCONC + int4( mmCONC )
                  if( mmCONC .le. minCONC ) minCONC = mmCONC
                  if( mmCONC .ge. maxCONC ) maxCONC = mmCONC

* calculate distributions
                  if( CONDIT .eq. '.' ) iGOOD    = iGOOD    + 1
                  if( CONDIT .eq. 'S' ) iSCALED  = iSCALED  + 1
                  if( CONDIT .eq. 'B' ) iDEBOND  = iDEBOND  + 1
                  if( CONDIT .eq. 'X' ) iDELAMIN = iDELAMIN + 1

             ELSE

                  iCORRUPT = iCORRUPT + 1

             ENDIF

* calculate sum of dielectric constant of asphalt
             IF ( permASP .gt. 0.0 .and. permASP .le. 12. ) THEN
                  sumDC = sumDC + permASP
             ELSE
                  largeDC = largeDC + 1
             ENDIF

* save conditions for plot ( PLOT.MTO ) opened in MTO90.for
            XLoc  = XLoc + wheel
            Mwave = Mwave + 1
            write ( 40, '(a,1x,f7.1,1x,i5)') CONDIT, XLOC, MWAVE

550     CONTINUE
       iaveC    = 0
       iaveD    = 0
       i10Wave  = 0
       iLL      = 0

      ENDIF

* for ANALOG data update XM-buffer counters
      IF ( mode .EQ. 0 ) THEN
          icount=icount+nbufsize                    ! update counter

* copy current XM buffer content to XMBUFNUM(1)
         if((icount.ge.limit).and.(iflag.eq.0))then
            left = int(float(ixmbuf(ixm)) - float(limit))
            STATUS=ALCB(XMBUFNUM(ixm),XMBUFNUM(1),0,0,left)
            iflag=1
            icount=0
         endif
      ENDIF

* get next CYCLE
500     CONTINUE

* finish all XM-buffers/or data yet, if not get more
400      IXM = IXM + 1
        IF ( ( IXM .LE. iLASTXM .AND. MODE .EQ. 0 ) .OR.
     +       ( IXM .LE. NCYCLE  .AND. MODE. EQ. 1 ) ) GO TO 130

        if ( isave .eq. 1 ) then
             if ( mode .eq. 1 )close ( 60 )
             close ( 70 )
             close ( 80 )
        endif

*  TERMINATE ATLAB OPERATIONS
        if ( mode. eq. 0 ) then
             STATUS = ALSTOP ()
             STATUS = ALTERM ()
        endif

*********************************
* calculate AVERAGEd parameters *
*********************************
* calculate AVERAGE asphalt thickness and concrete cover
        if ( Nwave .ne. icorrupt ) then
             iAVEASP  = int2( isumASP /int4( NWAVE - iCORRUPT ) )
             iAVECONC = int2( isumCONC/int4( NWAVE - iCORRUPT ) )
        else
             iAVEASP  = 999
             iAVECONC = 999
             minASP   = 999
             maxASP   = 999
             minCONC  = 999
             maxCONC  = 999
        endif

* calculate distribution in %
        GOOD     = float( iGOOD     ) * 100. / float( nwave )
        SCALED   = float( iSCALED   ) * 100. / float( nwave )
        DELAMIN  = float( iDELAMIN  ) * 100. / float( nwave )
        DEBOND   = float( iDEBOND   ) * 100. / float( nwave )
        CORRUPT  = float( iCORRUPT  ) * 100. / float( nwave )

        DAMAGED = SCALED + DELAMIN + DEBOND
        TOTAL   = GOOD + SCALED + DELAMIN + DEBOND + CORRUPT

* calculate average dielectric constant of asphalt
        if ( Nwave .ne. LargeDC .and. iaveASP .ne. 999 ) then
             ASPdc = sumdc/float( Nwave - largedc )
        else
             write(*,*)' [warning] asphalt dielectric is very large'
             ASPdc = 999.
        endif

* save processed results (TABLE.MTO)  opened in MTO90.for
        write ( 30 ) iaveASP, iaveCONC, GOOD, SCALED, DELAMIN, DEBOND,
     +                CORRUPT, DAMAGED, TOTAL, ASPdc

* output LINE SUMMARY to printer/monitor
      OPEN ( 80, file = 'PRN')
      write(80,250)title
```

```
250    format(1H1,1x,a60)
       write(*, 210) LINE
       write(80,210) LINE
210    format(1x,/,' * LINE # ',i2,' SUMMARY *')
       if ( wheel .eq. 0.10 ) then
         cmwheel = wheel * 100.
         write(*, 212) CMWheel
         write(80,212) CMWheel
212       format(1x,'[Fifth wheel]',5x,'= ',f4.1,' cm',1x,\)
       else
         write(*, 214) Wheel
         write(80,214) Wheel
214       format(1x,'[Fifth wheel]',5x,'= ',f4.1,' m',1x,\)
       endif
       write(*, 216) FPAMP
       write(80,216) FPAMP
216    format(5x,' [Flat Plate]',5x,'= ',f10.6,' v')
       write(*, 218) ASPdc
       write(80,218) ASPdc
218    format(1x,'[dielectric constant] of asphalt = ',f10.6,' averaged')
       if ( iconcrete .eq. 1 ) then
           write(*, 220) ASPHALT, CONCRETE
           write(80,220) ASPHALT, CONCRETE
220         format(1x,'[Core Samples (mm)]',2x,'ASPHALT = ',f5.1,
     +              5x,'CONCRETE = ',f5.1)
       else
           write(*, 222) ASPHALT
           write(80,222) ASPHALT
222         format(1x,'[Core Samples (mm)]',2x,'ASPHALT = ',f5.1,
     +              5x,'CONCRETE = no input')
       endif
       write(*, 224) minASP, maxASP, iaveASP
       write(80,224) minASP, maxASP, iaveASP
224    format(1x,'[Asphalt (mm)]  ',5x,'min: ',i3,5x,'maxi: ',i3,
     +                               5x,'AVE = ',i3)
       write(*, 226) minCONC, maxCONC, iaveCONC
       write(80,226) minCONC, maxCONC, iaveCONC
226    format(1x,'[Concrete (mm)] ',5x,'min: ',i3,5x,'maxi: ',i3,
     +                               5x,'AVE = ',i3,/)

       write(*, 228) SCALED
       write(80,228) SCALED
228    format(1x,'[ DAMAGED area ]',5x,'SCALED       : ',f6.2,' %')
       write(*, 230) DELAMIN
       write(80,230) DELAMIN
230    format(1x,'                ',5x,'DELAMINATED  : ',f6.2,' %')
       write(*, 232) DEBOND
       write(80,232) DEBOND
232    format(1x,'                ',5x,'DEBONDED     : ',f6.2,' %')
       write(*, 234)
       write(80,234)
234    format(1x,'                ',5x,23('-'))
       write(*, 236) DAMAGED
       write(80,236) DAMAGED
236    format(1x,'                ',5x,'DAMAGED area : ',f6.2,' %',/)
```

```

      write(*, 238)DAMAGED
      write(80,238)DAMAGED
238    format(1x,'[ CONDITIONS    ]',5x,'DAMAGED area : ',f6.2,' %')
      write(*, 240) GOOD
      write(80,240) GOOD
240    format(1x,'                ',5x,'GOOD    area : ',f6.2,' %')
      write(*, 242) CORRUPT
      write(80,242) CORRUPT
242    format(1x,'                ',5x,'CORRUPT data : ',f6.2,' %')
      write(*, 246)
      write(80,246)
246    format(1x,'                ',5x,23('-'))
      write(*, 248) TOTAL
      write(80,248) TOTAL
248    format(1x,'                ',5x,'TOTAL        : ',f6.2,' %')
      close ( 80 )

      RETURN
      END
```

```
$STORAGE:2

* locate preliminary peaks C and D

* fl/c peakCD90.for

      SUBROUTINE PEAKCD90 ( fpamp, maxpka, ASPHALT, ICONCRETE, CONCRETE,
     +                      LocC, LocD)

      parameter ( NBUFSIZE = 512  )  ! total sample per raw trace
      parameter ( NOPT     = 325  )  ! total sample per corrected trace
      parameter ( Mcycle   = 126  )  ! max 126 waveforms per buffer
      parameter ( MaxLine  =  20  )  ! max 20 Lines per site
      parameter ( dcASP    = 6.2  )  ! estimated DC of asphalt
      parameter ( dcCONC   = 8.22 )  ! estimated DC of concrete
      parameter ( drange   = -0.2 )  ! peak D threshold level

      common /BLK1/ ref(NBUFSIZE), ydata(NBUFSIZE), zdata(NBUFSIZE)

******************
* Locate PEAK A *    surface reflection  (air - asphalt interface)
******************
      LocA  = maxpka
      peakA = ydata ( LocA )

*******************
* check BAD DATA *
*******************
* test for corrupted data.
* data set is good for processing if the ratio of fpamp-peakA to fpamp
* is within the bound of 15% to 75%.
      test = ( fpamp - peakA ) * 100./fpamp
      IF ( test .gt. 75 .or. test .lt. 15 ) GO TO 100

******************
* Locate PEAK B *    transient peak
******************
      peakB = 1.0
      LocB  = LocA + 10       ! default LocB value
      DO J  = LocA, LocA + 15
            if ( ydata(j) .le. peakB ) then
                  peakB = ydata(j)
                  LocB  = j
            endif
      ENDDO

******************************
*  estimate PEAK C WINDOW   *   calculate IEDGE
* using ASPHALT CORE SAMPLE *
******************************
      Cwindow = ( asphalt * ( dcASP**0.5 ) )/6.
      IEDGE   = LocA + int( Cwindow )

******************
* Locate PEAK C *     asphalt to concrete interface
```

```
*****************
      locLOW = IEDGE - 5     ! 5 samples to the left
      locHGH = IEDGE + 5     ! 5 samples to the right
      if ( locLOW .le. LocB ) locLOW = LocB      ! LocC cannot be < LocB
      if ( locHGH .ge. nopt ) locHGH = nopt      ! LocC cannot be > Nopt
      LocC  = locLOW
      peakC = -10.0
      DO J = locLOW, locHGH
         if ( ydata(j) .ge. peakC ) then
            peakC = ydata(j)
            LocC  = j
         endif
      ENDDO

* search LocC to the LEFT of the window if LocC = locLOW
      IF ( LocC .eq. locLOW ) THEN
         iflag = 0
         j     = locLOW
10       continue
         if( ( ydata(j) .ge. ydata(j-1) ) .and.
     +        ( ydata(j) .ge. ydata(j-2) ) ) then
            LocC  = j      ! get LocC
            iflag = 1
         endif
         j = j - 1
         if ( j .le. LocB ) go to 15
         if ( iflag .eq. 0 ) go to 10
15     continue
      ENDIF

* search LocC to the RIGHT of the window if LocC = locHGH
      IF ( LocC .eq. locHGH ) THEN
         iflag = 0
         j     = locHGH
20       continue
         if ( ( ydata(j) .ge. ydata(j+1) ) .and.
     +         ( ydata(j) .ge. ydata(j+2) ) ) then
            LocC  = j      ! get Loc C
            iflag = 1
         endif
         j = j + 1
         if ( j .ge. nopt ) go to 25
         if ( iflag .eq. 0 ) go to 20
25       continue
      ENDIF
      peakC = ydata( LocC )

* REMARK for peakC location :
* normally, locC is some value between locLOW < locC < locHGH
* if locC = locLOW or locC = locHGH, check the Cwindow parameter
* and adjust the width accordingly.

************************
* get ready for PEAK D *
************************
```

```
* locate the first valley after locC for operation of locD
        locLOW = LocC
        locHGH = LocC + 15
        if ( locHGH .ge. nopt ) locHGH = nopt
        minC  = locLOW
        iflag = 0
        do j = locLOW, locHGH
              IF ( iflag .eq. 0 ) THEN
                   if ( ( ydata(j) .le. ydata(j+1) ) .and.
     +                  ( ydata(j) .le. ydata(j+2) ) ) then
                       iflag = 1
                       minC  = j
                   endif
              ENDIF
        enddo

********************************
*   estimate PEAK D window    *     calculate IEDGE
* using CONCRETE CORE SAMPLES *
********************************
        IF ( ICONCRETE .EQ. 1 ) THEN
             Dwindow  = ( concrete * ( dcCONC**0.5 ) )/6.
             IEDGE    = LocC + int( Dwindow )
             locLOW   = IEDGE - 5
             locHGH   = IEDGE + 5
        ENDIF

******************************
*  estimate peak D window   *
* generated by the computer *
******************************
        IF ( locLOW .le. minC .or. ICONCRETE .EQ. 0 ) THEN
             locLOW = minC
             locHGH = minC + 30
        ENDIF

******************
* Locate PEAK D *   rebar reflections
******************
        IXW = 0      ! clear flag for EXTENDED window
 30     continue
        if ( locHGH .ge  nopt ) locHGH = nopt
        LocD  = locLOW
        peakD = -10.
        do j = locLOW, locHGH
              if( ydata(j) .ge. peakD ) then
                   peakD = ydata(j)
                   LocD  = j
              endif
        enddo

* peak D amplitude is too low, search again
        if ( peakD .le. drange .and. IXW .eq. 0 ) then
             locHGH = locHGH + 10
             if ( locHGH .ge. nopt ) IXW = 1    ! out of bound
```

```
          go to 30
      endif

* peak D = peak C, search again
      IF ( LocD .le. LocC ) THEN
          locLOW = minC
          locHGH = minC + 30
          if ( minC .eq. LocC ) then
              locLOW = LocC + 10
              locHGH = LocC + 30
          endif
          if ( locHGH .ge. nopt ) locHGH = nopt
          peakD = -10.
          do j = locLOW, locHGH
               if ( ydata(j) .ge. peakD ) then
                   peakD = ydata(j)
                   LocD  = j
               endif
          enddo
      ENDIF

      GO To 300

***********************
*      report        *
* corrupted waveforms *
***********************
100   CONTINUE
      LocA    = 999
      LocB    = 999
      LocC    = 999
      LocD    = 999

300   CONTINUE

      RETURN
      END

```

```
$STORAGE:2

* locate peaks A,B,C and D
* measure asphalt and concrete cover
* identify scaling, delamination, debonding, good

* fl/c PEAKS90.for

      subroutine PEAKS90 ( fpamp, iwave, itrace, i10Wave, i10count,
     +mxC1, mxD1, iaveC, iaveD, LocC1, locC2, LocD1, LocD2, permASP,
     +isave )
      parameter ( NBUFSIZE = 512  )   ! total sample per raw trace
      parameter ( NOPT     = 325  )   ! total sample per corrected trace
      parameter ( Mcycle   = 126  )   ! max 126 waveforms per buffer
      parameter ( MaxLine  =  20  )   ! max 20 Lines per site
      parameter ( dcASP    = 6.2  )   ! estimated DC of asphalt
      parameter ( dcCONC   = 8.22 )   ! estimated DC of concrete
      parameter ( drange   = -0.2 )   ! peak D threshold level

      character CONdit

      common /BLK1/ ref(NBUFSIZE), ydata(NBUFSIZE), zdata(NBUFSIZE)
      common /BLK2/ NUMCYCLE( MAXLINE ), mmASP, mmCONC, CONDIT, ASPdc

*****************
* Locate PEAK A *   surface reflection  (air - asphalt interface)
*****************
      LocA = 130
      peakA = 0.
      do j = 1,nopt
            if( ydata(j) .ge. peakA ) then
                LocA = j
                peakA = ydata(j)
            endif
      enddo

*****************
* TEST BAD data *
*****************
* test for corrupted data.
* data set is good for processing if the ratio of fpamp-peakA to fpamp
* is within the bound of 15% to 75%.
      test = ( fpamp - peakA ) * 100./fpamp
      IF ( test .gt. 75 .or. test .lt. 15 ) GO TO 100

*****************
* Locate PEAK B *    transient peak
*****************
      peakB = 1.0
      LocB  = LocA + 10       ! default LocB value
      DO J  = LocA, LocA + 15
            if ( ydata(j) .le. peakB ) then
                peakB = ydata(j)
                LocB  = j
            endif
      ENDDO

*******************
* R1 RATIO = B/A *
*******************
      RONE = abs( peakB/peakA )

******************
* Locate PEAK C *   use iaveC, LocC1, LocC2
******************
      locLOW = iaveC - LocC1    ! 5 samples to the left
      locHGH = iaveC + LocC2    ! 5 samples to the right
      if ( locLOW .le. LocB ) locLOW = LocB      ! LocC cannot be < LocB
      if ( locHGH .ge. nopt ) locHGH = nopt      ! LocC cannot be > Nopt
      LocC  = locLOW
      peakC = -10.0
      DO J = locLOW, locHGH
            if ( ydata(j) .ge. peakC ) then
                  peakC = ydata(j)
                  LocC  = j
            endif
      ENDDO

* search LocC to the LEFT of the window if LocC = locLOW
      IF ( LocC .eq. locLOW ) THEN
            iflag = 0
            j     = locLOW
10          continue
            if( ( ydata(j) .ge. ydata(j-1) ) .and.
     +          ( ydata(j) .ge. ydata(j-2) ) ) then
                LocC  = j      ! get LocC
                iflag = 1
            endif
            j = j - 1
            if ( j .le. LocB ) go to 15
            if ( iflag .eq. 0 ) go to 10
15        continue
      ENDIF

* search LocC to the RIGHT of the window if LocC = locHGH
      IF ( LocC .eq. locHGH ) THEN
            iflag = 0
            j     = locHGH
20          continue
            if ( ( ydata(j) .ge. ydata(j+1) ) .and.
     +            ( ydata(j) .ge. ydata(j+2) ) ) then
                  LocC  = j     ! get Loc C
                  iflag = 1
            endif
            j = j + 1
            if ( j .ge. nopt ) go to 25
            if ( iflag .eq. 0 ) go to 20
25          continue
      ENDIF
      peakC = ydata( LocC )
```

```

* REMARK for peakC location :
* normally, locC is some value between locLOW < locC < locHGH
* if locC = locLOW or locC = locHGH, check the Cwindow parameter
* and adjust the width accordingly.

******************
* R2 ratio = C/A *
******************
      RTWO = abs( peakC/peakA )

************************
* dielectric constant *
*     of ASPHALT      *
************************
      rho     = ( peakA/fpamp ) * (-1.0)
      permASP = ( (1-rho)*(1-rho) )/( (1+rho)*(1+rho) )

**********************
* ASPHALT depth (mm) *
**********************
      mmASP = int( ( 6.0 * float( LocC-LocA ) ) / permASP**0.5 )

************************
* get ready for PEAK D *
************************
* locate the first valley after locC for operation of locD
      locLOW = LocC
      locHGH = LocC + 15
      if ( locHGH .ge. nopt ) locHGH = nopt
      minC  = locLOW
      iflag = 0
      do j = locLOW, locHGH
            IF ( iflag .eq. 0 ) THEN
                  if ( ( ydata(j) .le. ydata(j+1) ) .and.
     +                  ( ydata(j) .le. ydata(j+2) ) ) then
                        iflag = 1
                        minC  = j
                  endif
            ENDIF
      enddo

*****************
* Locate PEAK D *    use iaveD, LocD1, LocD2
*****************
      locLOW = iaveD - LocD1
      locHGH = iaveD + LocD2

      IF ( locLOW .le. minC ) THEN
            locLOW = minC
            locHGH = minC + 30
      ENDIF

      if ( locHGH .ge. nopt ) locHGH = nopt
      LocD  = locLOW
      peakD = -10.
      do j = locLOW, locHGH
             if( ydata(j) .ge. peakD ) then
                 peakD = ydata(j)
                 LocD  = j
             endif
      enddo

* peak D = peak C, search again
      IF ( LocD .eq. LocLOW .or. LocD .eq. LocHGH ) THEN
            locLOW = LocD - 5
            locHGH = LocD + 5
            if ( locLOW .le. minC ) then
                 locLOW = minC
                 locHGH = minC + 30
            endif
            if ( locHGH .ge. nopt ) locHGH = nopt
            peakD = -10.
            do j = locLOW, locHGH
                   if ( ydata(j) .ge. peakD ) then
                        peakD = ydata(j)
                        LocD  = j
                   endif
            enddo
      ENDIF
      if ( LocD .le. LocC ) then
            LocLOW = minC
            LocHGH = minC + 30
            if ( minC .eq. LocC )then
                 LocLow = LocC+10
                 LocHGH = LocLow + 30
            endif
            if ( locHGH .ge. nopt ) LocHGH = nopt
            peakD= -10.
            do J = LocLOW, locHGH
                   if (ydata(j).ge.peakD)then
                       peakD=ydata(j)
                       LocD = j
                   endif
            enddo
         endif

* peakD LE 0.0
        if ( peakD .lt. 0.0 ) then
             iflag = 0
             J     = LocD
30           if ( ydata(J) .ge. ydata(J+1) .and. ydata(J) .gt. 0.0 )then
                  peakD = ydata(J)
                  LocD  = J
                  iflag = 1
            endif
            J = J+1
            if ( iflag .eq. 0 .and. J .le. nopt ) go to 30
        endif

```

```
*******************
* CONCRETE COVER *
*   thickness    *
*******************
      mmCONC = int( ( 6.0 * float( LocD-LocC ) )/ dcCONC**0.5 )

******************
*  count #s of  *
* ZERO CROSSING *
******************
          izero1 = 0
          izero2 = 1
          izero3 = 1
          izero4 = 1
          iZXing = 0

        DO 200 J = LocC + 1, LocD - 1
              zxing = ydata( j+1 ) - ydata( j )

* count zero crossing
              if ( zxing .lt.0 .and. izero1 .eq. 0 ) then
                  izero1 = 1
                  izero2 = 0
                  izxing = izxing + 1
                  go to 200
              endif

              if( zxing .ge. 0 .and. izero2 .eq. 0 ) then
                 izero2 = 1
                 izero3 = 0
                 izxing = izxing + 1
                 go to 200
              endif

              if ( zxing .lt. 0 .and. izero3 .eq. 0 ) then
                  izero3 = 1
                  izero4 = 0
                  izxing = izxing + 1
                  go to 200
              endif

              if ( zxing .ge. 0 .and. izero4 .eq. 0 ) then
                  izero4 = 1
                  izxing = izxing + 1
              endif
200   CONTINUE

***************    * Scaling       : RONE >= 0.32,  RTWO >= 0.15   icond = 10
*  identify  *    * Delamination : izxing = 2                    icond = 20
* conditions *    * Debonding    : RONE <  0.32,  RTWO >= 0.15   icond = 30
***************    * Good          : izxing = 4                    icond = 40
* set thresholds R1 and R2
      R1 = 0.32
      R2 = 0.15          ! XT-based was 0.28

```

```
* Scaling = 'S'
     IF ( RONE .ge. R1 .and. RTWO .ge. R2 ) THEN
          CONDit = 'S'
          icond  = 10          ! for debug.mto file

* Delamination = 'X'
     ELSE if ( IZXING .le. 2 ) then
          CONDit = 'X'
          icond  = 20

* Debonding = 'B'
     ELSE if ( RONE .lt. R1 .and. RTWO .ge. R2 ) then
          CONDit = 'B'
          icond  = 30

* Good = '.'
     ELSE if ( IZXING .gt. 2 ) then
          CONDit = '.'
          icond  = 40

     ENDIF
     GO TO 300

***********************
*       report        *
* corrupted waveforms *
***********************
100   CONTINUE
     LocA    = 999
     LocB    = 999
     LocC    = 999
     LocD    = 999

     RONE    =  9.99999
     RTWO    =  9.99999
     mmASP   = 999
     mmCONC  = 999
     IZXING  =  9
     CONDit  = '*'
     permASP = 999.

     peakB   = 999.
     peakC   = 999.
     peakD   = 999.

300   CONTINUE

*********
* debug *  write processed data to DEBUG.MTO opened in DART90.for
*********
     if ( isave .eq. 1 ) then

     kwave = iwave - i10Wave + i10count
     ktrace = itrace - i10Wave + i10count
     if( ktrace .le. 0 ) ktrace = ktrace + Mcycle
```

```

      write( 20, 40 ) kwave, ktrace, LocA, LocB, LocC, LocD,
     +mmASP, mmCONC, IZXING, ICOND, permASP, peakA, peakB, peakC,
     +peakD, iaveC, iaveD, mxC1, mxD1, LocC1, LocC2, LocD1, LocD2,
     +RONE, RTWO
 40   format(1x,I5,1x,I3,2x,4(I3,1x),1x,2(I3,1x),1x,I1,1x,i2,2x,f10.6,2x
     +,4(f10.6,1x),1x,2(i3,1x),1x,2(i3,1x),1x,4(i3,1x),2(f10.6,1x))

      endif

      RETURN
      END

```

```
$storage:2

* FL/c TABLE90.for

      SUBROUTINE TABLE90 (TITLE, NUMLINE, FPAMP, ASPHALT, ICONCRETE,
     +                    CONCRETE, WHEEL)

      PARAMETER ( MAXLINE  = 20  )
      PARAMETER ( MaxLEN   = 100 )  ! max output 100m per time

      CHARACTER TITLE*60, clearscreen*4
      INTEGER*2 oneASP, oneCONC, aveASP, aveCONC

      common /BLK2/ NUMCYCLE( MAXLINE ), mmASP, mmCONC, CONDIT, ASPdc

      DIMENSION oneASP( MaxLEN, MAXLINE ), oneCONC( MaxLEN, MAXLINE )
      DIMENSION aveASP( MAXLINE ), aveCONC( MAXLINE ), dcASP( MAXLINE )
      DIMENSION GOOD( MAXLINE ), CORRUPT( MAXLINE ), DAMAGED( MAXLINE )
      DIMENSION SCALED( MAXLINE ), DELAMIN( MAXLINE ), DEBOND( MAXLINE )
      DIMENSION TOTAL( MAXLINE )
      CLEARSCREEN = char(27)//'['//'2'//'J'

      OPEN ( 10, file = 'PRN' )

*=======================================================
*| FILE = SUMMARY.MTO ( 20, UNKNOWN, FORMATTED, WRITE ) |
*| open/close                                           |
*=======================================================
      OPEN ( 20, file = 'SUMMARY.MTO',status='unknown',form='formatted')
          rewind( 20 )

* calculate average bridge length over NUMLINE run
      sum = 0
      DO  I  = 1, NUMLINE
            Blength  = float( NUMCYCLE( i ) ) * wheel
            sum      = sum + Blength
      ENDDO
      BLENGTH  = sum/float( NUMLINE )
      Clength  = Blength               ! save bridge length

* adjust bridge length/fifth wheel for print out
      IF ( wheel .ne. 0.10 ) Clength = Clength/wheel

* set counters for fifth wheel
      Ibeg = 1
      Iend = int( Clength )
      IF ( Iend  .gt. MaxLen ) Iend = MaxLen

* write to printer and SUMMARY.MTO opened in THKTABLE.for
      write ( 10, '(1H1,/,1x,a/)' ) ' [ SUMMARY ]'
      write ( 20, '(1x,/,1x,a/)'  ) ' [ SUMMARY ]'

      CALL GETDAT ( iyr, imon, iday )
      CALL GETTIM ( ihr, imin, isec, i100th )
      write ( 10, 5 ) imon,iday,iyr,ihr,imin,isec,i100th
      write ( 20 ,5 ) imon,iday,iyr,ihr,imin,isec,i100th
 5     format(1x,'DATE = ',i2,1x,i2,'.',i4,10x,'TIME = ',i2,3(':',i2))

      write ( 10, '(1x,/,1x,a/)' ) TITLE
      write ( 20, '(1x,/,1x,a/)' ) TITLE

      write ( 10, 10 ) BLENGTH, NUMLINE
      write ( 20, 10 ) BLENGTH, NUMLINE
 10    format(1x,'AVERAGed Bridge Length = ',f6.1,' metre ',
     +        5x 'NUMBER of LINES processed = ',i2,/)

      write ( 10, 20 ) ( I, I = 1, NUMLINE )
      write ( 20, 20 ) ( I, I = 1, NUMLINE )
 20    format(1x,'metre',5x,i2,19(11x,i2))
      write ( 10, 30 )
      write ( 20, 30 )
 30    format(8x,265('-'))

      LOC = 0        ! global counter for fifth wheel
 100   CONTINUE

*=====================================================================
*| FILE = TABLE.MTO ( 30, OLD, UNFORMATTED, READ, SIGNALAD.for) |
*| open/close                                                  |
*=====================================================================
* get the TABLE.MTO file
      OPEN ( 30,file='TABLE.MTO',status='OLD',form='unformatted' )
      rewind ( 30 )

      DO J = 1, NUMLINE

            Jend = int( float( numcycle(J) ) * wheel )
            if ( wheel .ne. 0.10 ) Jend = Jend/int(wheel)

            kk   = 0
            do K = 1, Jend

* read TABLE.MTO opened in THKTABLE.for
                  READ( 30 ) mmASP, mmCONC

                  if ( K .ge. Ibeg .and. K .le. Iend ) then
                       kk = kk + 1
                       oneASP ( kk, J ) = mmASP
                       oneCONC( kk, J ) = mmCONC
                  endif
            enddo

            if ( Jend .lt. Iend ) then
                 jj = Clength - Jend
                 do K = 1, jj
                       kk = kk + 1
                       oneASP ( kk, J ) = 999
                       oneCONC( kk, J ) = 999
                 enddo
            endif
```

```

* read TABLE.MTO opened in THKTABLE.for
              READ( 30 ) iaveASP, iaveCONC, aGOOD, aSCALED, aDELAMIN,
     +                      aDEBOND, aCORRUPT, aDAMAGED, aTOTAL, AdcASP

              aveASP ( J ) = iaveASP
              aveCONC( J ) = iaveCONC
              GOOD   ( J ) = aGOOD
              SCALED ( J ) = aSCALED
              DELAMIN( J ) = aDELAMIN
              DEBOND ( J ) = aDEBOND
              CORRUPT( J ) = aCORRUPT
              DAMAGED( J ) = aDAMAGED
              TOTAL  ( J ) = aTOTAL
              dcASP  ( J ) = AdcASP

        ENDDO
        CLOSE ( 30 )

        iFIFTH = 1                          ! set fifth wheel distance
        if ( wheel .eq. 10.0 ) iFIFTH = 10
        DO I = 1, kk
              LOC = LOC + iFIFTH
              write(10,40) LOC, (oneASP(i,j), oneCONC(i,j), j= 1,NUMLINE)
              write(20,40) LOC, (oneASP(i,j), oneCONC(i,j), j= 1,NUMLINE)
 40            format(1x,i5,2x,20(i4,1x,'(',i4,')',2x))
        ENDDO

        Ibeg = Iend + 1
        Imid = Clength - Ibeg + 1
        if ( Imid .gt. MaxLen ) Imid = MaxLen
        Iend = Ibeg + Imid - 1
        IF ( Ibeg .Le. CLength ) GO TO 100

* OVERALL averager parameters
        iaveASP  = 0
        iaveCONC = 0
        AdcASP   = 0.0
        aGOOD    = 0.0
        aSCALED  = 0.0
        aDELAMIN = 0.0
        aDEBOND  = 0.0
        aCORRUPT = 0.0
        aDAMAGED = 0.0
        aTOTAL   = 0.0

        DO I = 1, NUMLINE
              iaveASP  = iaveASP  + aveASP ( I )
              iaveCONC = iaveCONC + aveCONC( I )
              AdcASP   = AdcASP   + dcASP  ( I )
              aGOOD    = aGOOD    + GOOD   ( I )
              aSCALED  = aSCALED  + SCALED ( I )
              aDELAMIN = aDELAMIN + DELAMIN( I )
              aDEBOND  = aDEBOND  + DEBOND ( I )
              aCORRUPT = aCORRUPT + CORRUPT( I )
              aDAMAGED = aDAMAGED + DAMAGED( I )
              aTOTAL   = aTOTAL   + TOTAL  ( I )
        ENDDO

        iaveASP  = iaveASP  /        NUMLINE
        iaveCONC = iaveCONC /        NUMLINE
        AdcASP   = AdcASP   / float( NUMLINE )
        aGOOD    = aGOOD    / float( NUMLINE )
        aSCALED  = aSCALED  / float( NUMLINE )
        aDELAMIN = aDELAMIN / float( NUMLINE )
        aDEBOND  = aDEBOND  / float( NUMLINE )
        aCORRUPT = aCORRUPT / float( NUMLINE )
        aDAMAGED = aDAMAGED / float( NUMLINE )
        aTOTAL   = aTOTAL   / float( NUMLINE )

        write ( 10, '(1H1,/,1x,a)' ) ' [ SUMMARY ]'
        write ( 20, '(1x,/,1x,a)' ) ' [ SUMMARY ]'
        write ( 10, '(1x,/,1x,a/)' ) TITLE
        write ( 20, '(1x,/,1x,a/)' ) TITLE

        call GETTIM ( ihr, imin, isec, i100th )
        write ( 10, 5 ) imon,iday,iyr,ihr,imin,isec,i100th
        write ( 20 ,5 ) imon,iday,iyr,ihr,imin,isec,i100th

        write ( 10, 220 ) FPAMP
        write ( 20, 220 ) FPAMP
 220    format(1x,/,' [ FLAT PLATE AMPLITUDE ]  =  ',f10.6,' v')
        if ( wheel .eq. 0.10 ) then
             wheel = 10.0
             write ( 10, 230 ) WHEEL
             write ( 20, 230 ) WHEEL
 230         format(1x,'[ FIFTH WHEEL ]            =  ',f10.6,' cm')
        else
             write ( 10, 240 ) WHEEL
             write ( 20, 240 ) WHEEL
 240         format(1x,'[ FIFTH WHEEL            ]  =  ',f10.6,' m')
        endif
        write ( 10, 250 ) ASPHALT
        write ( 20, 250 ) ASPHALT
 250    format(1x,'[ ASPHALT  core sample ]  =  ',f5.1,' mm')
        if ( ICONCRETE .eq. 1 ) then
             write ( 10, 260 ) CONCRETE
             write ( 20, 260 ) CONCRETE
 260         format(1x,'[ CONCRETE core sample ]  =  ',f5.1,' mm',/)
        else
             write ( 10, 270 )
             write ( 20, 270 )
 270         format(1x,'[ CONCRETE core sample ]  =  NO INPUT',/)
        endif

        write ( 10, '(a/)' )' [ AVERAGed DIELECTRIC CONSTANT OF ASPHALT ]'
        write ( 20, '(a/)' )' [ AVERAGed DIELECTRIC CONSTANT OF ASPHALT ]'
        write ( 10, 120 ) ( I, I = 1, NUMLINE )
        write ( 20, 120 ) ( I, I = 1, NUMLINE )
        write ( 10, 60 )
```

```
      write ( 20, 60 )
60     format(1x,171('-'))
      write ( 10, 70 ) ( dcASP( i ), i = 1, numline )
      write ( 20, 70 ) ( dcASP( i ), i = 1, numline )
70     format(12x,20(2x,f6.2))

      write ( 10, '(1x,/,1x,a/)' ) ' [ AVERAGed ASPHALT THICKNESS ]'
      write ( 20, '(1x,/,1x,a/)' ) ' [ AVERAGed ASPHALT THICKNESS ]'
      write ( 10, 120 ) ( I, I = 1, NUMLINE )
      write ( 20, 120 ) ( I, I = 1, NUMLINE )
      write ( 10, 60 )
      write ( 20, 60 )
      write ( 10, 110 ) ( aveASP( i ), i = 1, numline )
      write ( 20, 110 ) ( aveASP( i ), i = 1, numline )
110    format(1x,'millimetres',3x,i4,19(4x,i4))
      write ( 10, '(1x,/,1x,a/)' ) ' [ AVERAGed CONCRETE COVER ]'
      write ( 20, '(1x,/,1x,a/)' ) ' [ AVERAGed CONCRETE COVER ]'
      write ( 10, 120 ) ( I, I = 1, NUMLINE )
      write ( 20, 120 ) ( I, I = 1, NUMLINE )
      write ( 10, 60 )
      write ( 20, 60 )
      write ( 10, 110 ) ( aveCONC( i ), i = 1, numline )
      write ( 20, 110 ) ( aveCONC( i ), i = 1, numline )

      write ( 10, '(1x,/,1x,a/)' ) ' [ PERCENTAGE OF DAMAGES ]'
      write ( 20, '(1x,/,1x,a/)' ) ' [ PERCENTAGE OF DAMAGES ]'
      write ( 10, 120 ) ( I, I = 1, NUMLINE )
      write ( 20, 120 ) ( I, I = 1, NUMLINE )
120    format(1x,15x,i2,19(6x,i2))
      write ( 10, 60 )
      write ( 20, 60 )
      write ( 10, 140 ) ( SCALED ( i ), i = 1, numline )
      write ( 20, 140 ) ( SCALED ( i ), i = 1, numline )
140    format(1x,'  SCALed     ',20(2x,f6.2))
      write ( 10, 150 ) ( DELAMIN( i ), i = 1, numline )
      write ( 20, 150 ) ( DELAMIN( i ), i = 1, numline )
150    format(1x,'DELAMINATed',20(2x,f6.2))
      write ( 10, 160 ) ( DEBOND ( i ), i = 1, numline )
      write ( 20, 150 ) ( DEBOND ( i ), i = 1, numline )
160    format(1x,' DEBONDed   ',20(2x,f6.2))
      write ( 10, 60 )
      write ( 20 ,60 )
      write ( 10, 170 ) ( DAMAGED( i ), i = 1, numline )
      write ( 20, 170 ) ( DAMAGED( i ), i = 1, numline )
170    format(1x,'  DAMAGED   ',20(2x,f6.2))

      write ( 10, '(1x,/,1x,a/)') '[ CONDITIONS ]'
      write ( 20, '(1x,/,1x,a/)') '[ CONDITIONS ]'
      write ( 10, 120 ) ( I, I = 1, NUMLINE )
      write ( 20, 120 ) ( I, I = 1, NUMLINE )
      write ( 10, 60 )
      write ( 20, 60 )
      write ( 10, 170 ) ( DAMAGED ( i ), i = 1, numline )
      write ( 20, 170 ) ( DAMAGED ( i ), i = 1, numline )
      write ( 10, 180 ) ( GOOD    ( i ), i = 1, numline )
      write ( 20, 180 ) ( GOOD    ( i ), i = 1, numline )
180    format(1x,'   GOOD      ',20(2x,f6.2))
      write ( 10, 190 ) ( CORRUPT( i ), i = 1, numline )
      write ( 20, 190 ) ( CORRUPT( i ), i = 1, numline )
190    format(1x,' CORRUPTed ',20(2x,f6.2))
      write ( 10, 60 )
      write ( 20, 60 )
      write ( 10, 210 ) ( TOTAL( i ), i = 1, numline )
      write ( 20, 210 ) ( TOTAL( i ), i = 1, numline )
210    format(1x,'  TOTAL     ',20(2x,f6.2))

* write overall results
      write ( *,*) clearscreen
      CALL GETTIM ( ihr, imin, isec, i100th )
      write ( *, 5 ) imon,iday,iyr,ihr,imin,isec,i100th
      write ( *, '(1x,a)' ) TITLE

      write ( 10, '(1x,/,1x,a/)' ) '[ OVERALL AVERAGES ]'
      write ( 20, '(1x,/,1x,a/)' ) '[ OVERALL AVERAGES ]'
      write (  *, '(1x,a)' ) '[ OVERALL AVERAGES ]'
      write ( 10, 280 ) AdcASP
      write ( 20, 280 ) AdcASP
      write (  *, 280 ) AdcASP
280    format(1x,'[ dielectric constant of aspahlt ] = ',f10.6,/)
      write ( 10, 290 ) iaveASP
      write ( 20, 290 ) iaveASP
      write (  *, 290 ) iaveASP
290    format(1x,'[ asphalt thickness (mm) ]          = ',i4,/)
      write ( 10, 310 ) iaveCONC
      write ( 20, 310 ) iaveCONC
      write (  *, 310 ) iaveCONC
310    format(1x,'[ concrete cover (mm) ]             = ',i4,/)
      write ( 10, '(a/)' ) ' [ PERCENTAGE of DAMAGES ]'
      write ( 20, '(a/)' ) ' [ PERCENTAGE of DAMAGES ]'
      write (  *, '(a)' ) ' [ PERCENTAGE of DAMAGES ]'
      write ( 10, 320 ) aSCALED
      write ( 20 ,320 ) aSCALED
      write (  * ,320 ) aSCALED
320    format(1x,'SCALED        ',f6.2,' %')
      write ( 10, 330 ) aDELAMIN
      write ( 20, 330 ) aDELAMIN
      write (  *, 330 ) aDELAMIN
330    format(1x,'DELAMINATED  ',f6.2,' %')
      write ( 10, 340 ) aDEBOND
      write ( 20, 340 ) aDEBOND
      write (  *, 340 ) aDEBOND
340    format(1x,'DEBONDED      ',f6.2,' %')
      write ( 10, 350 )
      write ( 20, 350 )
      write (  *, 350 )
350    format(1x,20('-'))
      write ( 10, 360 ) aDAMAGED
      write ( 20, 360 ) aDAMAGED
      write (  *, 360 ) aDAMAGED
360    format(1x,'DAMAGED       ',f6.2,' %',/)
```

```
      write ( 10, '(a/)' ) ' [ CONDITIONS ]'
      write ( 20, '(a/)' ) ' [ CONDITIONS ]'
      write (  *, '(a)' ) ' [ CONDITIONS ]'
      write ( 10, 370 ) aDAMAGED
      write ( 20, 370 ) aDAMAGED
      write (  *, 370 ) aDAMAGED
370    format(1x,'DAMAGED      ',f6.2,' %')
      write ( 10, 380 ) aGOOD
      write ( 20, 380 ) aGOOD
      write (  *, 380 ) aGOOD
380    format(1x,'GOOD         ',f6.2,' %')
      write ( 10, 390 ) aCORRUPT
      write ( 20, 390 ) aCORRUPT
      write (  *, 390 ) aCORRUPT
390    format(1x,'CORRUPTED    ',f6.2,' %')
      write ( 10, 350 )
      write ( 20, 350 )
      write (  *, 350 )
      write ( 10, 410 ) aTOTAL
      write ( 20, 410 ) aTOTAL
      write (  *, 410 ) aTOTAL
410    format(1x,'TOTAL        ',f6.2,' %')

      write ( 10,'(1H1)')
      write ( 10,'(1H1)')
      close ( 10 )
      close ( 20 )

      RETURN
      END
```

```
$storage:2

      program DECK90

* DECK90 is a software package which employs the commerical plotting
* library PLOTMS4.LIB to perform the required plotting functions.
* refer to the documentation for each routine's function.
*

      dimension disp(20)
      character name*15, xlabel*15, ylabel*15, title*70
      character charsx, charS, charB, charX, blank, bad
      character symbol(1000,20), rsym, sym, key
      integer*4 numcycle(20), total, ksym(20)
      integer line(20)

* variable definitons -
*     line(.)      : bridge line
*     numcycle(.) : number of radar traces per line
*     disp(.)      : displacement per line
*     symbol(.,.) : fault symbol of each line
*     ksym(.)      : trace counter of each line
*     charX        : X symbol for delamination
*     charS        : S symbol for scaling
*     charB        : B symbol for debonding
*     blank        : . symnbol for undamaged area
*     bad          : * symbol for corrupted data
*     rsym         : plot option identifier
*     sym          : temporary storage for symbols
*     total        : total number of radar waveform of the current plot
*     name         : reserved file name for PLOTMS.LIB
*     xlabel       : x-axis label
*     ylabel       : y-axis label
*     title        : plot title
*     key          : keyboard character

* initialization
      xlabel = 'm'
      ylabel = 'width (metre)'
      limit  = 1000    ! limit to 100m (1000 waveforms at 0.10m) per line

* read data file PLOT.mto
      open(10,file='PLOT.mto',status='old',form='formatted')
          rewind(10)
          read(10,1) NUMLINE, WHEEL
1         format(1x,i2,1x,f5.2)

* input displacement of each line
      write(*,2)
      write(*,3)
      write(*,4)numline
      write(*,5)
      do i=1,numline
           write(*,6)i
           read(*,*)disp(i)
```

```
        enddo
  2      format(1x,/,/,8x,'[ PLOT BRIDGE DECK CONDITIONS ]')
  3      format(1x,/,10x,'MAXImum input is 20 lines')
  4      format(9x,'- PLOT.mto has  ',i2,' LINE(s) -')
  5      format(1x,/,9x,'** input DISPlacement (m) ** ',/)
  6      format(7x,'line ',i2,' is..................... ',\)

* input plot selection
107    write(*,8)
        write(*,9)
        write(*,10)
        write(*,11)
        write(*,12)
        write(*,13)
        read(*,'(a)')rsym
        if(rsym.eq.'B'.or.rsym.eq.'S'.or.rsym.eq.'X'.or.rsym.eq.'A'.or.    &
       +rsym.eq.'b'.or.rsym.eq.'s'.or.rsym.eq.'x'.or.rsym.eq.'a')then
           go to 106
        else
           go to 107
          endif
106    continue
  8      format(1x,/,7x,'select plot option : ')
  9      format(15x,'for  SCALEing        type  S ')
  10     format(15x,'for  DEBONDing       type  B ')
  11     format(15x,'for  DELAMINation  type  X ')
  12     format(15x,'for  COMBINed plot type  A ')
  13     format(15x,'for  ABORT              press Cntrl C   ....',\)

* input plot title
         write(*,15)
         write(*,16)
         read(*,'(A)')title

* read data file PLOT.mto
         do i=1,numline
              read(10,17)LINE(i),NUMCYCLE(i)
         enddo
         CLOSE ( 10 )
  15      format(1x,/,13x,' - Label plot title -')
  16      format(1x,'-->',\)
  17      format(1x,i2,1x,i5)

* calculate bridge length
         total=0
         iover=0
         imax=0
         do i=1,numline
                    total=total+numcycle(i)
                    if( numcycle(i) .gt. limit ) iover = 1
                    if( numcycle(i) .ge. imax  ) imax  = numcycle(i)
         enddo

         LENGTH = INT( float(total) * wheel / float(numline) )
```

```
* report bridge length to user
      write(*,18) LENGTH
 18    format(1x,/,7x,'bridge LENGTH is .... ',i4,' metre')

* calculate upper bound for the width of the bridge deck layout
      dispm = 0.0
      do i = 1,numline
          dis = disp(i)
          if ( dis .ge. dispm ) dispm = dis
      enddo
      dis = int(dispm) + 1
      write(*,33)dis
      write(*,34)
      read(*,'(a)')key
      if(key.ne.' ')then
        write(*,35)
        read(*,*)dis
      endif
 33    format(1x,/,7x,'bridge WIDTH is .... ',f5.0,' metre')
 34    format(1x,/,7x,'<cr> if okay, other keys to alter width : ',\)
 35    format(1x,/,7x,'input new bridge width (m) : ',\)

* report number of plots to user
      ipaper = ( length/100 ) + 1
      if( length .gt. 100 )then
        write(*,19)ipaper
        write(*,20)
        write(*,21)ipaper
      endif
 19    format(1x,/,6x,'program [DECK90] will run ',i2,' plots')
 20    format(4x,'each plot covers 100m of bridge length')
 21    format(8x,'please have ',i2,' papers ready !')

* plot a new section here
 190   continue
      iend=0
      itime=1
      xpart1=0
      xpart2=length
      if(xpart2.gt.100)xpart2=100

 120   continue
      istart=1+iend
      iend=limit*itime
      if(istart.gt.imax)go to 121

* read the data file CONDIT.mto
      if(itime.eq.1)then
        write(*,22)
      else
        write(*,23)
      endif
 22    format(8x,'reading data file CONDIT.mto')
 23    format(1x,/,7x,'... reading MORE data ...')

```

```
      open(30,file='CONDIT.mio',status='old',form='formatted')
          rewind(30)
          do 101 i=1,numline
                 k=0
                 iread=numcycle(i)
                 do j=1,iread
                     read(30,'(a,1x,f7.1,1x,i5)') SYM, XLOC, NWAVE
                     if(j.ge.istart.and.j.le.iend)then
                        k=k+1
                        symbol(k,i)=sym
                     endif
                 enddo
                 ksym(i)=k
101          continue
      close(30)

* check plotter and get ready
      write(*,25)
108    write(*,26)
      read(*,'(a)')key
      if(key.ne.' ')go to 108
25     format(1x,/,16x,'Check PLOTTER')
26     format(1x,/,7x,'... strike RETurn when READY ...',\)

* report the bridge length of the current plot section
      if(itime.ne.1)then
         xpart1=xpart2
         xpart2=xpart1+100.
      endif
      xpart=xpart2
      if(xpart.gt.length)xpart=float(length)
      write(*,27)xpart1,xpart
27     format(1x,/,7x,'current plot section is from ',f7.2,' to ',
     +f7.2,' m')

*  set up X-coordinate parameters
        xstart=xpart1
        xmin=xpart1
        xmax=xpart2
        xorg=xmin
        xstep=0.5
        xdif=xpart2-xpart1
        if(xdif.gt.100.and.length.gt.100)xdif=100
        if(xdif.gt.40)xstep=1
        if(xdif.gt.80)xstep=2
        xtic=xdif/xstep
        ixstep=int(xtic)
        xmaj=2
        if(xdif.gt.40)xmaj=5*xstep
        xrange=xdif/10.

        xfact=0.05
        if(xdif.ge.10.and.xdif.le.30)xfact=0.075
        if(xdif.gt.30.and.xdif.le.60)xfact=0.15
        if(xdif.gt.60.)xfact=0.25
```

```

* set up Y-coordinate parameters
      ymin=0
      ymax=dis
      yorg=0
      ystep=1.
      iystep=int(ymax)-1
      ymaj=1
      yrange=(ymax-ymin)/4.
      if(yrange.lt.1)fact=0.015
      if(yrange.ge.1)fact=0.05
      if(yrange.ge.2)fact=0.1
      if(yrange.ge.3)fact=0.125

* set up HP plotter unit equivalent (check HP plotter manual)
* P1x=ix1, P2x=ix2, P1y=iy1, P2y=iy2
* set up User unit equivalent
* U1x=xbeg, U2x=xend, U1y=ybeg, U2y=yend
      name='scratch.dat'
      vel=16.
      ix1=0
      ix2=10300
      iy1=100
      iy2=7650

      xbeg=xmin-xrange
      xend=xmax+xrange
      ybeg=ymin-yrange
      yend=ymax+yrange

      x1=(ix2-ix1)/(xend-xbeg)
      x2=ix1-(xbeg*x1)
      y1=(iy2-iy1)/(yend-ybeg)
      y2=iy1-(ybeg*y1)

* call PLOTMS4.LIB routines
* initiate plotter
      call zinit(2,1,name)
      call zplot(ix1,ix2,iy1,iy2,xbeg,xend,ybeg,yend,xorg,yorg)
      call zlt(-1,0,0)
      call zsp(1)
      call zvs(vel)

* draw boundary and write labels
      call zxaxis(xorg,xmax,xmaj,0,1,0)
      call zyaxis(yorg,ymax,ymaj,0,1,0)
      call zptxtb(15,xlabel,1)
      call zptxtc(15,ylabel,1)
      call zptxta(70,title,1)
      call zpu

* draw vertical grid lines
       x=xmin
       y=ymin
       yii=(y1*ymax)+y2
```

```
        iiy=int(yii)
        xtic=xmaj/xstep
        ii=0
        do 105 i=1,ixstep
                x=x+xstep
                if(x.gt.xpart)go to 105
                xi=(x1*x)+x2
                ix=int(xi)
                yi=(y1*y)+y2
                ii=ii+1
                if(ii.eq.xtic)then
                   ii=0
                   yi=(y1*(y-fact))+y2
                endif
                iy=int(yi)
                call zpa(ix,iy)
                call zpd
                call zpa(ix,iiy)
                call zpu
105         continue

* draw horizontal grid lines
          xi=(x1*xmin)+x2
          ix=int(xi)
          xii=(x1*xpart)+x2
          iix=int(xii)

          do 109 i=1,numline
                  y=disp(i)
                  yi=(y1*y)+y2
                  iy=int(yi)
                  call zpa(ix,iy)
                  call zpd
                  call zpa(iix,iy)
                  call zpu
109         continue
          yi=(y1*dis)+y2
          iy=int(yi)
          call zpa(ix,iy)
          call zpd
          call zpa(iix,iy)
          call zpu

* identify plot selection
        isym=0
        if(rsym.eq.'X'.or.rsym.eq.'x')isym=1
        if(rsym.eq.'B'.or.rsym.eq.'b')isym=2
        if(rsym.eq.'S'.or.rsym.eq.'s')isym=3
        if(rsym.eq.'A'.or.rsym.eq.'a')isym=4

* simulate time delay
        do 123 i=1,50000
                nop=0
123      continue
```

```
          if(isym.eq.1)then
             write(*,28)
          endif
          if(isym.eq.2)then
             write(*,29)
          endif
          if(isym.eq.3)then
             write(*,30)
          endif
          if(isym.eq.4)then
             write(*,31)
          endif
28         format(1x,/,7x,'... plotting x - DELAMINation ...')
29         format(1x,/,7x,'... plotting B - DEBONDing ...')
30         format(1x,/,7x,'... plotting S - SCALing ... ')
31         format(1x,/,7x,'... plotting X - COMBINed syms ...')

* assign symbols for different fault conditions
          call zpd
          blank='.'
          bad='*'
          charsx='x'
          charS='S'
          charB='B'
          charX='X'

* start plotting each line
          do 110 i=1,numline

* calculate spot location
                y=disp(i)-fact
                yaxis=(y1*y)+y2
                iyaxis=int(yaxis)
                step=wheel
                x=xstart+step-xfact

                iread=ksym(i)
                do 111 j=1,iread

                      if(x.gt.length)go to 111

                      xaxis=(x1*x)+x2
                      ixaxis=int(xaxis)

* plot x - for delamination
                      if(isym.eq.1.and.symbol(j,i).eq.charX)then
                         call zptxt(1,charsx,1,ixaxis,iyaxis)
                      endif

* plot B - for debonding
                      if(isym.eq.2.and.symbol(j,i).eq.charB)then
                         call zptxt(1,charB,1,ixaxis,iyaxis)
                      endif

* plot S - for scaling
```

```
                     if(isym.eq.3.and.symbol(j,i).eq.charS)then
                        call zptxt(1,charS,1,ixaxis,iyaxis)
                     endif

* plot X - for combined symbols
                  if(isym.eq.4)then
                  if(symbol(j,i).ne.blank.and.symbol(j,i).ne.bad)then
                     call zptxt(1,charX,1,ixaxis,iyaxis)
                  endif
                  endif

* update spot location
                      x=x+step
111            continue
110     continue
       call zpu

* simulate time delay
       if(length.le.100)then
          do 122 i=1,50000
                  nop=0
122        continue
       endif

       itime=itime+1
       if(iover.eq.1)go to 120

121     continue
       call zfinis

* repeat questionnaries
116     write(*,32)
       read(*,'(a)')key
32      format(1x,/,7x,'<cr> for NEXT plot, <Q>-STOP : ',\)

       if(key.eq.'q'.or.key.eq.'Q')go to 117
       if(key.eq.' ')then
114     write(*,8)
       write(*,9)
       write(*,10)
       write(*,11)
       write(*,12)
       write(*,13)
       read(*,'(a)')rsym
       if(rsym.eq.'B'.or.rsym.eq.'S'.or.rsym.eq.'X'.or.rsym.eq.'A'.or. &
     +rsym.eq.'b'.or.rsym.eq.'s'.or.rsym.eq.'x'.or.rsym.eq.'a')then
        go to 115
      else
        go to 114
       endif
115    continue
       write(*,15)
       read(*,'(A)')title
       write(*,25)
118     write(*,26)
```

```
          read(*,'(a)')key
          if(key.ne.' ')go to 118
          go to 190
          else
          go to 116
          endif
    117     continue

          end
```